

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Nina Žakelj

Napredno iskanje nepremičnin

DIPLOMSKO DELO
NA UNIVERZITETNEM ŠTUDIJU

MENTOR: doc. dr. Dejan Lavbič

Ljubljana, 2014

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov diplomskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil \LaTeX .

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Pri iskanju nepremičnin nam obstoječe informacijske rešitve ponujajo že znane načine iskanja - omejevanje po lokaciji, ceni, velikosti, številu sob, opremi ipd. Manjkajo predvsem podatki iz drugih virov, ki bi jih lahko smiselno vključili in na ta način izboljšali iskalno uporabniško izkušnjo. Iskalniki, ki delujejo predvsem v tujini npr. upoštevajo tudi čas, potreben za prevoz v službo ter bližino posameznih, pogosto obiskanih lokacij. Pri iskanju primerne nepremičnine namreč lahko iskalnik upošteva še te omejitve. V okviru diplomske naloge najprej identificirajte iskalnike nepremičnin v Sloveniji ter podatkovne vire, ki bi jih bilo smiselno integrirati za izboljšavo trenutne uporabniške izkušnje. Razvijte tudi prototip naprednega iskalnika nepremičnin v slovenskem okolju, ki naj temelji na odprtokodnih tehnologijah in gostuje kot storitev v oblaku. Integracija različnih podatkovnih virov naj bo implementirana kot samodejna storitev, ki se ves čas izvaja.

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisana Nina Žakelj, z vpisno številko **63080189**, sem avtorica diplomskega dela z naslovom:

Napredno iskanje nepremičnin

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelala samostojno pod mentorstvom doc. dr. Dejana Lavbiča,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 15. junija 2014

Podpis avtorja:

Zahvalila bi se svojemu mentorju doc. dr. Dejanu Lavbiču, za vse nasvete, strokovno pomoč in korekten odnos pri izdelavi diplomske naloge.

Zelo sem hvaležna staršema, ki sta mi omogočila študij in ostalim družinskim članom za podporo in spodbudo v času študija.

Hvala tudi sošolcem, za lepe trenutke in pomoč pri učenju ter ga. Barbari Peternel za lektoriranje diplomske naloge.

Kazalo

Seznam kratic in simbolov

Povzetek

Abstract

| | | |
|----------|--|----------|
| 1 | Uvod | 1 |
| 2 | Podobni pristopi | 3 |
| 2.1 | AutNo | 4 |
| 2.2 | Zillow | 5 |
| 2.3 | Trulia | 6 |
| 3 | Uporabljena orodja in tehnologije | 9 |
| 3.1 | Eclipse | 9 |
| 3.2 | Google App Engine | 9 |
| 3.3 | Google Drive | 11 |
| 3.4 | Google Maps Javacript API | 11 |
| 3.4.1 | Storitev Geocoding | 12 |
| 3.4.2 | Storitev Directions | 12 |
| 3.4.3 | Knjižnica Places | 13 |
| 3.5 | OAuth | 14 |
| 3.6 | Luščenje spletnih podatkov | 15 |
| 3.6.1 | HtmlUnit | 16 |
| 3.6.2 | XPath | 16 |

| | | |
|----------|--|-----------|
| 4 | Predstavitev naprednega iskalnika nepremičnin | 17 |
| 4.1 | Prijava v aplikacijo | 19 |
| 4.2 | Iskanje in prikaz nepremičnin | 19 |
| 4.2.1 | Parametri iskanja | 19 |
| 4.2.2 | Prikaz rezultatov iskanja | 21 |
| 4.2.3 | Prikaz podatkov o nepremičnini | 22 |
| 4.2.4 | Priljubljene nepremičnine | 24 |
| 4.2.5 | Prikaz objektov | 25 |
| 4.2.6 | Prikaz poti | 25 |
| 5 | Implementacija iskalnika nepremičnin | 29 |
| 5.1 | Pridobivanje podatkov iz spleta | 29 |
| 5.2 | Shranjevanje podatkov | 33 |
| 5.3 | Uporabniški vmesnik | 37 |
| 5.3.1 | Prijava v aplikacijo | 38 |
| 5.3.2 | Iskanje nepremičnin ter prikaz nepremičnin in objektov | 38 |
| 5.3.3 | Shranjevanje priljubljenih nepremičnin | 44 |
| 5.3.4 | Prikaz poti | 45 |
| 6 | Zaključek | 47 |

Slike

| | | |
|------|---|----|
| 2.1 | Iskalnik nepremičnin AutNo. | 4 |
| 2.2 | Iskalnik nepremičnin Zillow. | 6 |
| 2.3 | Iskalnik nepremičnin Trulia. | 7 |
| 3.1 | Pregled statistike izvajanja aplikacije v administracijski konzoli. | 11 |
| 4.1 | Diagram komponent aplikacije. | 18 |
| 4.2 | Zaslonska maska prijave v aplikacijo. | 19 |
| 4.3 | Zaslonska maska glavne strani aplikacije. | 20 |
| 4.4 | Prikaz rezultatov ob uspešnem iskanju. | 22 |
| 4.5 | Zaznamka za prikaz nepremičnin. | 22 |
| 4.6 | Okno za prikaz osnovnih podatkov o nepremičnini. | 23 |
| 4.7 | Okno za prikaz dodatnih podatkov o nepremičnini. | 24 |
| 4.8 | Zaznamka za prikaz priljubljenih nepremičnin. | 25 |
| 4.9 | Zaznamki za prikaz dodatnih objektov. | 26 |
| 4.10 | Okno za prikaz podatkov o objektih. | 26 |
| 4.11 | Prikaz poti med dvema lokacijama. | 27 |
| 5.1 | Diagram komponent sistema za avtomatsko pridobivanje podatkov. | 30 |
| 5.2 | Preglednice s tabelami in njihovimi stolpci. | 36 |

Primeri

| | | |
|------|---|----|
| 5.1 | Koda za pridobitev podatkov o lekarnah. | 30 |
| 5.2 | HTTP zahteva za pridobitev geografskih koordinat naslova. . . | 31 |
| 5.3 | Del rezultata HTTP zahteve na Primeru 5.2. | 31 |
| 5.4 | Dokument cron.xml v aplikaciji. | 33 |
| 5.5 | Programska koda za dostop do Google preglednic z Oauth 2.0. | 37 |
| 5.6 | Programska koda ki upravljanja s prijavo uporabnika. | 38 |
| 5.7 | Programska koda za inicializacijo Google zemljevida. | 39 |
| 5.8 | Programska koda za pridobitev in prikaz vrtcev na zemljevidu. | 41 |
| 5.9 | Podatki o vrtcu v JSON formatu. | 42 |
| 5.10 | AJAX klic za shranjevanje priljubljenih nepremičnin. | 44 |
| 5.11 | Programska koda za prikaz poti med točkama. | 46 |

Seznam kratic in simbolov

AJAX (angl. Asynchronous JavaScript and XML) - je skupina povezanih spletnih razvojnih tehnik, ki se na strani uporabnika uporabljajo za izdelavo asinhronih spletnih aplikacij.

API (angl. Application programming interface) - aplikacijski programski vmesnik, ki določa, kako bi morale posamezne komponente programske opreme sodelovati med seboj.

CSS (angl. Cascading Style Sheets) - slogovni jezik, ki se uporablja za opis videza in oblikovanje spletnih dokumentov napisanih v označevalnem jeziku.

EE (angl. Enterprise Edition)

HTML (angl. HyperText Markup Language) - je standardni označevalni jezik za oblikovanje spletnih strani.

HTTP (angl. Hypertext Transfer Protocol) - aplikacijski protokol, ki je temelj za izmenjavo podatkov v svetovnem spletu.

HTTPS (angl. Hypertext Transfer Protocol Secure) - komunikacijski protokol za varno izmenjavo podatkov v računalniškem omrežju.

IDE (angl. Integrated Development Environment) - integrirano razvojno okolje.

Java - objektno usmerjeni programski jezik.

JavaScript - objektni skriptni programski jezik.

jQuery - je JavaScript knjižnica ki omogoča enostavnejše upravljanje s HTML dokumenti.

OpenId - odprt standard, ki uporabnikom omogoča prijavo v več različnih, nepovezanih spletnih strani, ne da bi se pri tem registrirali.

XML (angl. Extensible Markup Language) - je označevalni jezik, ki določa format za opisovanje strukturiranih podatkov ali arhitektura za prenos podatkov, in njihovo izmenjavo med več omrežij.

Povzetek

Glavni cilj diplomske naloge je bil razviti spletno aplikacijo za iskanje nepremičnin, ki združuje podatke iz različnih spletnih virov. Aplikacija poleg iskanja po osnovnih kriterijih nepremičnine, omogoča tudi iskanje glede na oddaljenost od objektov kot so vrtec, osnovna šola, srednja šola, visokošolski zavod, zdravstvena ustanova, lekarna, železniška ali avtobusna postaja, poštna poslovalnica ter gasilska postaja, prikaz nepremičnin na Google zemljevidu in prikaz poti med dvema lokacijama. Uporabnik se lahko v aplikacijo prijavi z Google računom in si shrani priljubljene nepremičnine.

V začetnem delu diplomske naloge so predstavljeni podobni, že obstoječi, iskalniki nepremičnin na slovenskem področju in v tujini. Osrednji del diplomske naloge predstavlja opis uporabljenih orodij, tehnologij in konceptov ter natančen opis delovanja in implementacije vseh funkcionalnosti, ki jih izdelana aplikacija ponuja.

Ključne besede: nepremičnina, Google App Engine, Google Maps, luščenje spletnih podatkov

Abstract

The main objective of the thesis was to develop an online real estate search application that combines data from different web sources. In addition to the search by the basic real estate criteria, the application also allows search by distance from facilities such as kindergarten, primary school, secondary school, higher education institute, health facility, pharmacy, train or bus station, post office and fire station, display of property locations on Google Maps and driving directions between two locations. Users can also login to the application by using their Google accounts and save their favorite search results.

At the beginning of the thesis similar currently available real estate search engines in Slovenia and abroad are introduced. The main part of the thesis gives a description of all tools, technologies and concepts used, a detailed description of the application's functionality and a description of implementation of all functions provided in the application.

Ključne besede: real estate, Google App Engine, Google Maps, web scraping

Poglavje 1

Uvod

Nakup ali najem nepremičnine, ne glede na njen tip, predstavlja enega izmed bolj pomembnih dogodkov v življenju vsakega posameznika. Prav tako je lahko to zapleten proces, v katerem je potrebno sprejemati številne odločitve. Pri tem se lahko stori veliko napak, ki nato vplivajo na življenje vsakega posameznika in njegovih bližnjih. Zato se je potrebno iskanja nepremičnin lotiti preudarno in si pri tem pridobiti čim več podatkov.

Pri iskanju primernih nepremičnin nam v današnjem času veliko pomaga svetovni splet, kjer se lahko najde številne informacije. Za iskanje informacij o nepremičninah na trgu so na voljo različne informacijske rešitve, ki omogočajo iskanje po nepremičninah po različnih kriterijih in vsebujejo informacije o karakteristikah same nepremičnine in minimalne informacije o okolici nepremičnine. Vse ostale podatke, kot so na primer podatki o samem kraju, bližina različnih objektov kot so šola, trgovine in podobno, možnost javnega prevoza pa mora posameznik posebej ročno poiskati na spletu.

Ideja te diplomske naloge je identificirati, kateri podatkovni viri so za uporabnika najbolj pomembni pri izbiri nepremičnine in tako izdelati spletno aplikacijo za iskanje nepremičnin, ki bo združevala te podatkovne vire in s tem uporabniku olajšala iskanje. Aplikacija bo izdelana na brezplačnih orodjih za izdelavo spletnih aplikacij in bo uporabljala odprtokodne in brezplačne tehnologije, zato bo imela določene omejitve v zmogljivostih delovanja.

V nadaljevanju diplomske naloge so najprej predstavljeni že obstoječi iskalniki nepremičnin v slovenskem okolju in opisane tri spletne aplikacije, ki vsebujejo podobne funkcionalnosti, kot jih ponuja izdelana aplikacija, ki se uporabljajo za iskanje nepremičnin v tujini. V tretjem poglavju so opisana vsa orodja, tehnologije in pristopi, ki so bili uporabljeni pri izdelavi aplikacije, oziroma jih aplikacija uporablja pri svojem delovanju. Vse funkcionalnosti, ki jih aplikacija ponuja in podrobno delovanje aplikacije, je opisano v četrtem poglavju. V petem poglavju sledi podroben opis implementacije aplikacije, tudi s primeri programske kode. V zadnjem, šestem poglavju, so predstavljene še zaključne ugotovitve, opis težav pri razvoju aplikacije in predlogi za izboljšavo in nadgradnjo aplikacije.

Poglavje 2

Podobni pristopi

Na svetovnem spletu obstaja veliko iskalnikov nepremičnin, ki poleg iskanja po osnovnih informacijah o nepremičnini kot so lokacija, mikrolokacija, cena, velikost, število sob ipd. ponujajo iskanje tudi po drugih podatkih, kot so na primer bližina javnega prevoza ter prikazujejo druge zanimive podatke o okolici nepremičnine (bližina šol, različne statistike okolice). Vendar pa se ti iskalniki pojavljajo predvsem za nepremičnine v tujini.

V slovenskem prostoru sicer obstaja kar nekaj večjih iskalnikov nepremičnin kot so na primer bolha.com [1], Nepremicnine.net [2], GoHome.si [3], salomon.si [4], Nepremicninar.com [5] ter javni vpogled v nepremičnine na Geodetski upravi Republike Slovenije [6], vendar med njimi le iskalnik Nepremicnine.net, ponuja še dodatne funkcionalnosti poleg osnovnih podatkov o nepremičnini. Iskalnik Nepremicnine.net prikaže tudi lokacijo nepremičnine na zemljevidu in omogoča informativni izračun kredita za izbrano nepremičnino.

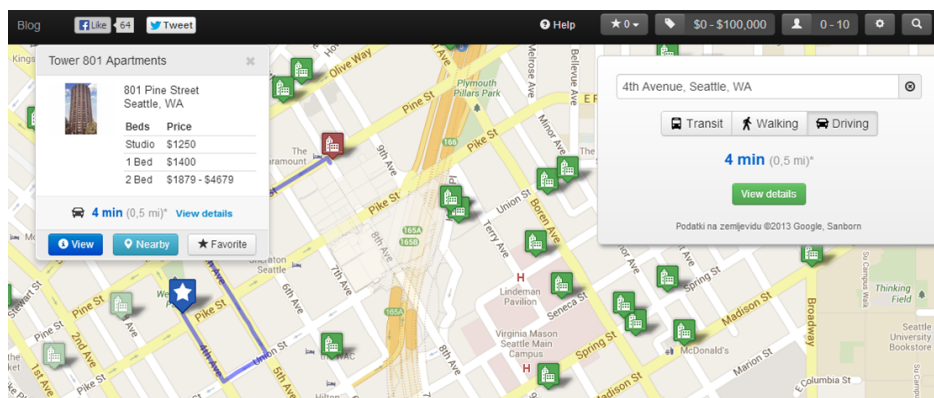
Obstajajo tudi manjši iskalniki, ki vsebujejo zelo majhno število nepremičnin, med katerimi sta najdena dva iskalnika Domart [7] in Hiša nepremičnine [8], ki omogočata tudi funkcionalnosti, ki jih bo vsebovala izdelana aplikacija. Domart omogoča iskanje nepremičnin tudi glede na lokacijo na zemljevidu, Hiša nepremičnine pa omogoča prikaz poti od izbrane nepremičnine do poljubnega naslova.

Ker v slovenskem okolju še ni spletnega iskalnika nepremičnin, ki bi tako

kot je ideja našega iskalnika poleg osnovnega iskanja združeval še več dodatnih funkcionalnosti in različne zanimive informacije o sami nepremičnini ali okolici nepremičnine, bodo v nadaljevanju predstavljeni trije iskalniki, ki to ponujajo, vendar omogočajo iskanje nepremičnin le v Združenih državah Amerike.

2.1 AutNo

AutNo je prvi iskalnik stanovanj v Združenih državah Amerike, ki omogoča ljudem, ki se vsak dan vozijo v službo in so brez avtomobila, najti stanovanje v bližini javnega prevoza ali večjih glavnih cest [9]. Sistem s pomočjo podatkov o linijah in voznih redih javnih prevozov prikaže nepremičnine, od katerih je čas za prevoz v službo čim krajši.



Slika 2.1: Iskalnik nepremičnin AutNo.

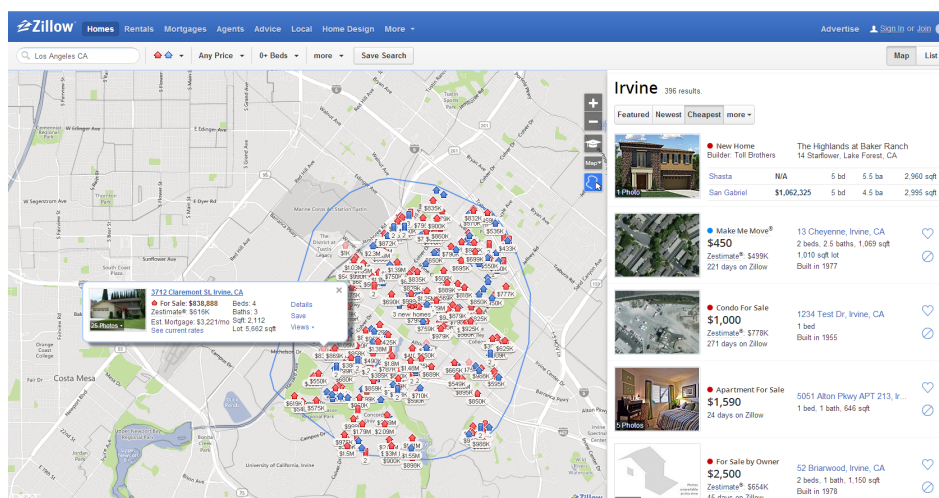
AutNo [10] je enostaven za uporabo, saj na začetku uporabnik vpiše le pošto številko (zip code) ali ime kraja, kjer želi stanovati ter lokacijo, kamor se vsak dan vozi (na primer lokacija službe). Nato se prikaže Google Map zemljevid, ki je prikazan na Sliki 2.1, z lokacijami stanovanj, ki so na voljo. Barva zaznamka za lokacijo stanovanja je odvisna od tega ali se stanovanje nahaja blizu avtobusne postaje, linije vlaka ali postaje podzemne železnice.

S klikom na zaznamek na zemljevidu, iskalnik prikaže osnovne podatke stanovanja, ter samodejno oceni koliko časa bi potrebovali od stanovanja do prej izbrane lokacije. V posebnem oknu se prikaže podatek o oddaljenosti nepremičnine od lokacije službe in ocenjen čas za vožnjo v službo. Uporabnik lahko izbira med tremi načini prevoza v službo: javni prevoz, vožnja z osebnim avtomobilom ali hojo. Za izbrani način iskalnik prikaže podrobno razčlenitev najkrajše poti do službe.

2.2 Zillow

Trenutno najbolj popularen iskalnik v Združenih državah Amerike [11]. Ima podatke o 100 milijonih nepremičnin čez celotne Združene države Amerike, tudi o nepremičninah ki še niso na trgu. Iskalnik Zillow [12] ponuja različne podatke o nepremičninah med katerimi so tudi spremembe cene nepremičnine v določenem časovnem obdobju (na primer v enem, petih ali desetih letih), zračni pogled na nepremičnino, cene primerljivih nepremičnin v izbranem okolju, stroški dodatnih priključkov, telefona, televizije, interneta, alarmnega sistema, podatek o tem ali je bil kakšen del nepremičnine v kratkem prenovljen, podatek o šolah, ki se nahajajo v bližini z oceno kakovosti posamezne šole.

Uporabnik na začetku poda mesto, sosesko ali naslov, kjer želi iskati nepremičnino. Zadetki nepremičnin se nato prikažejo na zemljevidu ter tudi kot seznam, ki se ga lahko razvršča po različnih kriterijih (zadnje dodani, cena, število sob, število kopalnic, leto izgradnje in podobno). Prikaz rezultatov iskalnika je viden na Sliki 2.2. Uporabnik lahko iskanje zoži z izbiro dodatnih parametrov kot so cena, število sob, število kopalnic, tip stanovanja, velikost, leto izgradnje, in iskanjem s ključnimi besedami, prav tako pa lahko z risanjem regije na zemljevidu zmanjša tudi samo območje iskanja. S klikom na zaznamek na zemljevidu se prikažejo osnovne informacije o nepremičnini, klik na podrobnosti pa uporabniku prikaže vse prej omenjene informacije o nepremičnini.



Slika 2.2: Iskalnik nepremičnin Zillow.

Če se uporabnik prijavi s svojim facebook računom, ali si kreira nov brezplačni račun v Zillow iskalniku, si lahko shrani različna iskanja in priljubljene nepremičnine.

2.3 Trulia

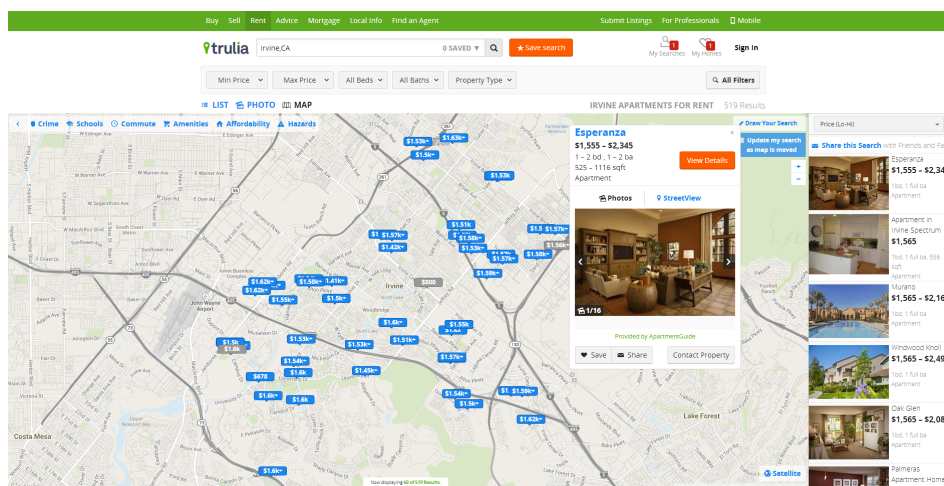
Tako kot prej omenjena iskalnika tudi spletna stran Trulia [13] predstavlja iskalnik nepremičnin v Združenih državah Amerike. Ponuja veliko različnih funkcionalnosti pri iskanju ustrezne nepremičnine in podatkov o nepremični.

Tudi pri tem iskalniku uporabnik najprej poda mesto v katerem želi iskati nepremičnino. Zadetke nepremičnin se lahko prikaže na tri načine: kot seznam nepremičnin po naslovih nepremičnine, kot seznam s slikami nepremičnin ali kot zaznamki na zemljevidu in seznam ob strani. Najbolj zanimiv je prikaz nepremičnin na zemljevidu, ki je viden tudi na Sliki 2.3, saj si uporabnik lahko na zemljevidu prikaže tudi statistiko kriminala v tem mestu, osnovne, srednje šole in fakultete glede na oceno šole, linije javnega prevoza in oddaljenost od podanega naslova (na primer lokacija službe), bližnje restavracije, banke, trgovine in bencinske črpalke, višina najemnine, cene

nepremičnine na posameznem območju mesta, možnost pojava različnih naravnih nesreč. Omogoča tudi izbiro velikosti območja z risanjem regije na zemljevid.

S klikom na zaznamek na zemljevidu se prikažejo osnovne informacije o nepremični, klik na podrobnosti pa uporabniku prikaže podrobne podatke o nepremičnini kot so cena, naslov, število sob, število kopalnic, dodatne lastnosti, podatki o soseski: bližnje šole, postaje javnega prometa.

Uporabnik si lahko kreira nov brezplačni račun ali pa se prijavi s svojim facebook računom, kar mu omogoča, da si shrani poizvedbo iskanja ter nepremičnine, ki so mu bile všeč.



Slika 2.3: Iskalnik nepremičnin Trulia.

Poglavje 3

Uporabljena orodja in tehnologije

3.1 Eclipse

Za izdelavo aplikacije je bilo uporabljeno orodje Eclipse Java EE IDE verzija Juno. Orodje je integrirano razvojno okolje, ki vsebuje osnovno delovno okolje in omogoča dodajanje različnih razširitev za prilagajanje tega okolja [14].

Uporabljen je Google razširitev za Eclipse [15], ki vsebuje razvojna orodja, ki omogočajo hitro načrtovanje, izdelavo, optimizacijo in nalaganje¹ App Engine aplikacij v oblak ter enostaven uvoz Google vmesnikov, ki jih izdelana aplikacija uporablja. Orodje omogoča razvoj aplikacij v mnogih programskih jezikih med katerimi so tudi C, C++, Java, JavaScript, PHP in Ruby.

3.2 Google App Engine

Google App Engine (GAE ali enostavno App Engine) [16] je platforma kot storitev² računalništva v oblaku, ki omogoča implementacijo in izvajanje aplikacij na Googlovi infrastrukturi. GAE aplikacijo je enostavno izdelati

¹angl. deploy

²angl. Platform as a service - PaaS

in vzdrževati ter ni potrebno skrbeti za vzdrževanje strežnikov, kar so tudi najpomembnejši razlogi, za izdelavo aplikacije v tem okolju. GAE prav tako omogoča enostavno nalaganje aplikacij na strežnike, kjer tečejo v varnem okolju³ in integracijo z drugimi Google storitvami v oblaku in Google vmesniki (API) [17].

Osnovni, brezplačni paket razvijalcu omogoča 500MB prostora za shranjevanje in dovolj moči centralne procesne enote⁴ ter dovolj pasovne širine, da je omogočeno 5 milijonov ogledov strani na mesec. Prav tako si razvijalec lahko izbere ali se bo njegova aplikacija izvajala na brezplačni domeni appspot.com, ali katerikoli drugi zasebni domeni. Če izbere brezplačno domeno, lahko na strežnik naloži do deset različnih aplikacij. Za vsako aplikacijo se kreira spletni naslov `http://<identifikator-aplikacije>.appspot.com`, kjer `<identifikator-aplikacije>` predstavlja enolični identifikator aplikacije, ki se določi ob kreiranju aplikacije. Razvijalcu je omogočeno tudi, da omeji dostop do aplikacije na le uporabnike znotraj njegove organizacije ali pa jo deli s celotnim svetom [18].

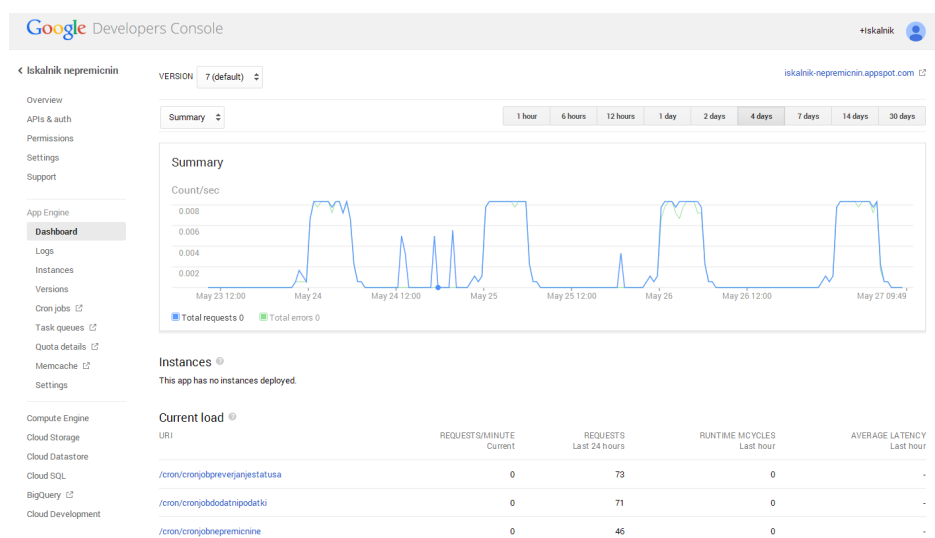
Aplikacije so lahko napisane v programskih jezikih Java, Python, PHP ali Go. Sama sem se odločila za programski jezik Java, saj App Engine omogoča uporabo standardnih javanskih tehnologij.

GAE omogoča tudi nadzor nad aplikacijo z administracijsko konzolo⁵ [20], ko je aplikacija že naložena v oblak in je v produkciji. To je posebna spletna stran, ki omogoča upravljanje z aplikacijo, nadzorovanje njenega izvajanja in spreminjanje nastavitev aplikacije. Aplikacijo je možno onemogočiti ali izbrisati, omogoča nastavljanje katera verzija aplikacije je trenutno v uporabi, pregledovanje napak ki so se pojavile pri dostopu do aplikacije, analizo prometa, različne statistike in še mnogo drugih informacij ter upravljanje z načrtovanimi opravili v aplikaciji [19]. Prikaz statistike izvajanja aplikacije v administracijski konzoli je prikazan na Sliki 3.1.

³angl. secure hosting environment

⁴angl. central processing unit

⁵administration console



Slika 3.1: Pregled statistike izvajanja aplikacije v administracijski konzoli.

3.3 Google Drive

Google Drive [21] je storitev v oblaku, ki uporabniku omogoča brezplačen 15Gb prostor za shranjevanje, urejanje, izmenjavo in dostop do datotek kot so slike, preglednice, dokumenti, predstavitve in podobno. V izdelani aplikaciji se Google Drive uporablja za bazo podatkov, kjer so podatki o nepremičninah, vrtcih, osnovnih šolah, srednjih šolah, visokošolski zavodih, lekarnah in zdravstvu shranjeni v javnih tabelah preglednic.

3.4 Google Maps Javascript API

Vmesnik Google Maps Javascript [22] omogoča prikaz Googlovega zemljevida v aplikacijah in razvijalcem ponuja različne knjižnice in druge tehnologije za prilagajanje zemljevida potrebam njihovih aplikacij. Tako se lahko na zemljevidu na primer prikaže lokacijo podanega naslova, bližnje trgovine, lokale, avtobusne postaje, bolnišnice in druge ustanove, pot od ene lokacije do druge z različnim načinom prevoza, ulični pogled in še mnogo drugih

stvari.

Ker največji del izdelane aplikacije temelji na prikazu nepremičnin in ostalih objektov na zemljevidu je Google Maps API najbolj pomemben vmesnik aplikacije in je omogočil izvedbo večino idej v aplikaciji.

V okviru tega vmesnika je uporabljena storitev⁶ Geocoding [23] za postavitev Zemljevida na izbran kraj in prikaz lokacij posameznih nepremičnin, storitev Directions [24] za prikaz poti od začetne do končne lokacije na zemljevidu in besedni opis poti, knjižnica Places [25] za iskanje bližnjih železniških in avtobusnih postaj, poštne poslovalnice ter gasilskih domov.

3.4.1 Storitev Geocoding

Geokodiranje⁷ je postopek pretvarjanja naslovov (primer Tržaška cesta 25, Ljubljana) v geografske koordinate (primer pasovna širina 46.044899 in pasovna dolžina 14.489231). Te koordinate se lahko potem uporabi za postavitev zaznamka⁸ na zemljevid. Obstaja tudi obratno geokodiranje⁹, ki pretvarja geografske koordinate v človeku berljive naslove.

Vse to omogoča tudi storitev Geocoding, in sicer preko objekta `google.maps.Geocoder` in metode `Geocoder.geocode()`, ki vrne rezultat v obliki `GeocoderResults` objekta, ali preko HTTP zahteve, ki lahko vrne rezultate v JSON ali XML formatu. V okviru razvite aplikacije se uporablja le drugi način, in sicer za določitev geografskih koordinat nepremičnin in ostalih podatkov v sklopu pridobivanja podatkov.

3.4.2 Storitev Directions

Storitev omogoča prikaz poti na zemljevidu od začetnega do končnega kraja, ki sta lahko podana bodisi kot besedna zveza oziroma naslov lokacije, ali kot pasovna širina in dolžina lokacije. Storitev lahko vrne več poti z različnimi

⁶angl. service

⁷angl. Geocoding

⁸angl. marker

⁹angl. Reverse Geocoding

vmesnimi točkami. Pot je na zemljevidu narisana kot krivulja med točkama obeh lokacij. Dodatno pa se lahko prikaže tudi besedni opis poti, ki se nahaja znotraj div HTML elementa.

3.4.3 Knjižnica Places

Knjižnica omogoča razvijalcem v aplikacijo vključiti iskanje lokacij različnih ustanov, geografskih lokacij ali pomembnih interesnih točk¹⁰, ki se nahajajo znotraj določenega območja, kot so meje na zemljevidu ali v območju okoli fiksne točke. V okviru knjižnice so omogočene štiri vrste iskanja:

- **Nearby search** (bližnje iskanje): omogoča iskanje lokacij glede na tip lokacije ali s ključnimi besedami znotraj določenega območja, ki je podan kot radij okoli geografske točke v metrih ali kot območje pravokotne oblike,
- **Radar search** (radar iskanje): omogoča iskanje lokacij s ključnimi besedami, s tipom lokacije ali z imenom lokacije znotraj določenega območja in vrne več rezultatov kot Nearby search ali Text search, vendar z manj podatki o lokaciji, le referenco na lokacijo ter pasovno širino in dolžino lokacije,
- **Text search** (iskanje s ključnimi besedami): vrne seznam lokacij z informacijami na podlagi podanega niza besed (na primer »prostovoljno gasilsko društvo«) in
- **Place details requests** (zahteva za podrobnosti lokacije): vrne bolj podrobne informacije o točno določeni lokaciji.

Rezultat vsakega iskanja je seznam objektov tipa `PlaceResult` [26], ki za lokacije vsebuje podatke o naslovu, pasovni širini in dolžini, nazivu, če gre za lokal tudi informacije o delovnem času in druge podobne informacije.

¹⁰angl. points of interest

Iskanja omogočajo tudi iskanje po tipu lokacije, ki podpira tipe [27] kot so na primer avtobusna postaja (`bus_station`), zdravnik (`doctor`), gasilska postaja (`fire_station`), bolnišnica (`hospital`) in še mnogo drugih.

3.5 OAuth

OAuth je odprtokodni standard za avtorizacijo, ki tretji osebi¹¹ omogoča pridobitev omejenega dostopa do HTTP storitve bodisi v imenu lastnika teh virov z dirigiranjem odobritve interakcije med lastnikom virov in HTTP storitvijo ali tako, da omogoči tretji osebi, da si pridobi dostop v svojem imenu, brez da bi pri tem razkrila svoje poverilnice¹² (uporabniško ime in geslo). Torej namesto, da bi se za dostop do zaščitениh virov uporabilo poverilnice, OAuth zagotavlja mehanizem, ki odobri dostop z uporabo žetona¹³ za dostop, ki je s soglasjem lastnika virov izdan s strani avtorizacijskega strežnika. Žeton določa, do katerih virov lahko tretja oseba dostopa in koliko časa ima za to. [28, 29].

Trenutno obstajata dve verziji standarda OAuth 1.0 in OAuth 2.0, ki jo Google uporablja za dostop do Google vmesnikov, in sicer v štirih korakih [30]:

- **Pridobitev OAuth 2.0 poverilnic:** že prej omenjena administracijska konzola za posamezno aplikacijo, omogoča tudi pridobitev identifikacije (`client ID`) in skrivnost odjemalca, ki sta znani aplikaciji in Google-u.
- **Pridobitev žetona** za dostop od Google avtorizacijskega strežnika (`Google Authorization Server`): preden lahko aplikacija z Google vmesniki dostopa do zasebnih podatkov, mora pridobiti žeton, ki dovoljuje dostop do tega vmesnika. En žeton lahko omogoča različne nivoje do-

¹¹angl. third-party

¹²angl. credentials

¹³angl. token

stopa do večih različnih vmesnikov. Seznam operacij in virov, ki jih žeton dovoljuje, upravlja parameter scope.

- **Pošiljanje žetona vmesniku:** ko aplikacija pridobi žeton za dostop ga v HTTP glavi pošlje Google vmesniku. Žetoni za dostop so veljavni le za seznam operacij in virov opisanih v parametru scope.
- **Osveževanje žetonov,** če je potrebno: žetoni imajo omejeno življenjsko dobo in aplikacija lahko po poteku te življenjske dobe pridobi osvežen žeton, ki nato aplikaciji omogoča pridobitev novega žetona za dostop.

3.6 Luščenje spletnih podatkov

Luščenje spletnih podatkov¹⁴ je proces pridobivanja nestrukturiranih in polstruktuiranih¹⁵ podatkov iz dokumentov, ki jih običajno predstavljajo spletne strani napisane v označevalnem jeziku kot sta HTML ali XHTML in analiza teh podatkov z namenom, da se iz njih izlušči določene podatke, ki se jih potem uporabi v drugem kontekstu [31].

Aplikacije, ki omogočajo luščenje spletnih podatkov, so programirane tako, da simulirajo človekovo raziskovanje svetovnega spleta in pridobivajo večje količine podatkov, ki bi bili drugače človeku težko dosegljive. Luščenje spletnih podatkov je zelo podobno spletnemu indeksiranju¹⁶, ki z uporabo spletnega robota¹⁷ indeksira informacije na spletu in je tudi splošna tehnika, ki je uporabljena v večini spletnih iskalnikov. Razlika med njima je le to, da se spletno indeksiranje uporablja bolj za izboljšanje učinkovitosti spletnih iskalnikov, luščenje spletnih podatkov pa za shranjevanje pridobljenih podatkov v bazo, ki so potem na voljo za poljubno uporabo [32].

Na spletu obstaja veliko različnih knjižnic, ki omogočajo razvijalcem izdelati programe za luščenje spletnih podatkov, v različnih programskih jezikih.

¹⁴angl. web scraping

¹⁵angl. semi-structured

¹⁶angl. web indexing

¹⁷angl. Internet bot, web crawler

Primeri teh knjižnic so: JSoup [33], HtmlUnit [34], node.io [35], Beautiful Soup [36], Html Agility Pack [37]. V izdelani aplikaciji se uporablja knjižnica HtmlUnit.

3.6.1 HtmlUnit

HtmlUnit je brskalnik brez grafičnega uporabniškega vmesnika, ki razvijalcu omogoča manipulacijo s spletnimi stranmi s pomočjo programov napisanih v programskem jeziku Java. HtmlUnit [34] zagotavlja vmesnik, ki omogoča klicanje¹⁸ spletnih strani, izpolnjevanje in pošiljanje obrazcev, klikanje po spletni strani, pridobivanje podatkov, ter tudi podporo za strani na osnovi JavaScript in delo z zapletenimi Ajax knjižnicami. Zagotavlja torej vse, kar tudi današnji brskalniki, in vključuje tudi možnost simulacije brskalnika Firefox ali Internet Explorer. Običajno se uporablja za simulacijo brskalnikov za potrebo samodejnega testiranja spletnih strani, uporaben pa je tudi za luščenje podatkov s spletnih strani.

Za dostop do posameznega HTML elementa se lahko uporabi eno izmed metod, ki vrnejo elemente glede na identifikator ali ime elementa, lahko pa se uporabi tudi XPath. V izdelani aplikaciji se uporabljata oba omenjena načina.

3.6.2 XPath

XPath [38] oziroma XML Path Language je poizvedovalni jezik, katerega glavni namen je navigacija preko XML dokumenta in naslavljanje oziroma iskanje različnih informacij v XML dokumentu. Za izbiro oziroma dostop do elementa v XML dokumentu uporablja izrazne poti, ki so podobne kot pri delu s tradicionalnim računalniškim datotečnim sistemom.

¹⁸angl. invoke

Poglavje 4

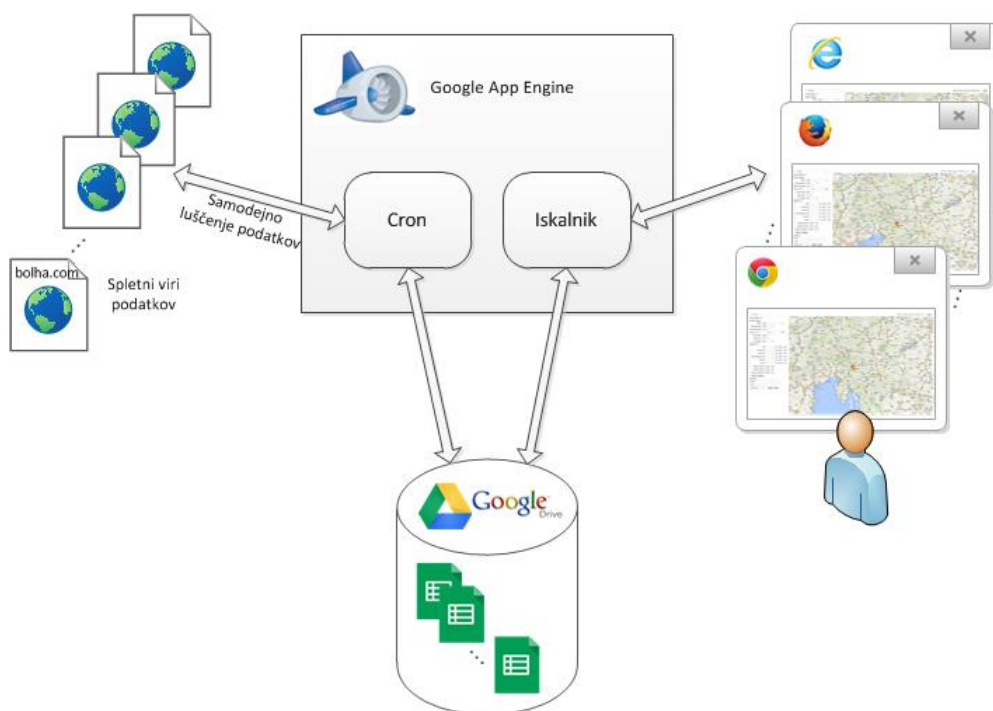
Predstavitev naprednega iskalnika nepremičnin

Namen te diplomske naloge je bil izdelati iskalnik nepremičnin, ki ponuja uporabniku več funkcionalnosti pri iskanju nepremičnin, kot to ponujajo trenutno obstoječi iskalniki v slovenskem okolju. Na izbiro funkcionalnosti, ki jih ponuja izdelan iskalnik, so vplivali tudi iskalniki, ki se uporabljajo za iskanje nepremičnin v tujini in so opisani v drugem poglavju. Funkcionalnosti, ki jih ponuja izdelan iskalnik so:

- Prijava v iskalnik z Google računom
- Iskanje nepremičnin po parametrih, ter tudi po oddaljenosti od izbranih objektov, med katerimi so vrtec, osnovna šola, srednja šola, visokošolski zavod, zdravstvena ustanova, lekarna, železniška in avtobusna postaja, poštna poslovalnica ter gasilska postaja
- Prikaz rezultatov iskanja na zemljevidu
- Prikaz dodatnih objektov na zemljevidu
- Prikaz poti od izbrane začetne lokacije do končne lokacije (na primer od izbrane nepremičnine do službe)
- Prikaz dodatnih podatkov o nepremičnini

- Shranjevanje priljubljenih nepremičnin

Sistem, v katerem se izvaja izdelana aplikacija, je sestavljen iz več komponent, ki so prikazane na diagramu na Sliki 4.1. Središče tega sistema je aplikacijski strežnik Google App Engine. Tu se nahaja vsa logika za izvajanje aplikacije, ki je razdeljena na dva dela. Prvi del je sam iskalnik nepremičnin, ki ga uporabnik vidi kot spletno aplikacijo na poljubnem brskalniku. Drugi del pa predstavljajo cron opravila, ki se samodejno izvajajo ponoči in pridobivajo podatke iz drugih spletnih virov, kot je na primer bolha.com in jih shranjujejo v podatkovno bazo sistema. Podatkovno bazo v sistemu predstavljajo preglednice v Google Drive, ki shranjujejo vse podatke, ki so pridobljeni iz spletnih virov in se uporabljajo v iskalniku.



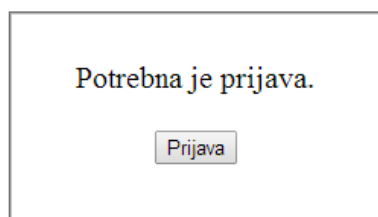
Slika 4.1: Diagram komponent aplikacije.

V nadaljevanju poglavja bodo vse omenjene funkcionalnosti podrobneje predstavljene.

4.1 Prijava v aplikacijo

Za dostop v aplikacijo je potrebna prijava z Google računom. Prijava uporabnikov je potrebna le zato, da si iskalnik lahko zapolni nepremičnine, ki si jih uporabnik izbere za priljubljene, kar bo natančneje opisano v nadaljevanju poglavja. Ko uporabnik odpre spletno stran

<http://iskalnik-nepremicnin.appspot.com>, kjer se nahaja iskalnik, se v primeru, da uporabnik še ni prijavljen z računom prikaže zaslonska maska na Sliki 4.2. Ko uporabnik klikne na gumb Prijava, ga aplikacija preusmeri na Google-ovo stran za prijavo z Google računom. Po prijavi, aplikacija uporabnika preusmeri na glavno stran iskalnika, ki je prikazana na Sliki 4.3. Večino strani predstavlja zemljevid, katerega center je postavljen na sredino območja Slovenije, na levi strani pa se nahaja prostor, ki zagotavlja ostale funkcionalnosti iskalnika.



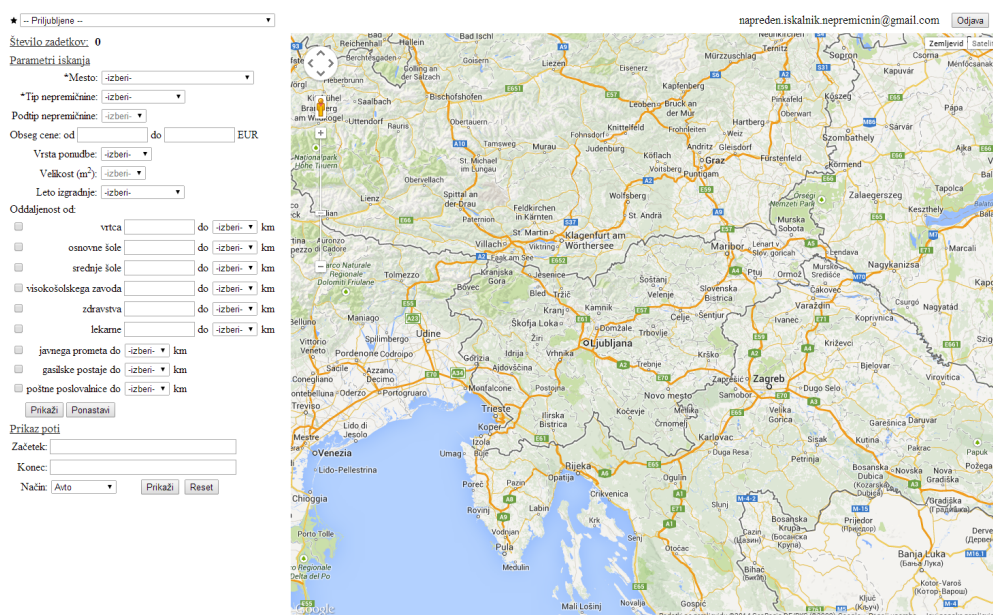
Slika 4.2: Zaslonska maska prijave v aplikacijo.

Če je uporabnik že prijavljen, se aplikacija odpre na glavni strani iskalnika. Uporabnik se lahko odjavi iz aplikacije s klikom na gumb Odjavi, poleg svojega elektronskega naslova zgoraj desno.

4.2 Iskanje in prikaz nepremičnin

4.2.1 Parametri iskanja

Uporabnik lahko nepremičnine išče po naslednjih kriterijih:



Slika 4.3: Zaslonska maska glavne strani aplikacije.

- **Mesto** - na izbiro so imena vseh občin, ki se nahajajo v Sloveniji.
- **Tip nepremičnine** - izbira se lahko med tipi: stanovanje, hiša, počitniški objekt, posest, garaža, poslovni prostor.
- **Podtip nepremičnine** - na izbiro so različni podtipi glede na to, kater tip nepremičnine je izbran.
- **Obseg cene** - uporabnik lahko vpiše interval želene cene.
- **Vrsta ponudbe** - način ponudbe nepremičnine. Izbira se lahko med načini prodam, oddam in podarim.
- **Velikost** - velikost nepremičnine v kvadratnih metrih. Na voljo je izbira intervalov velikosti nepremičnine, ki so različni glede na izbran tip nepremičnine.
- **Leto izgradnje** - leto, ko je bila nepremičnina zgrajena. Na voljo je izbira različnih intervalov leta izgradnje nepremičnine.

- **Oddaljenost** od vrtca, osnovne šole, srednje šole, visokošolskega zavoda, zdravstvene ustanove, lekarne, postaje javnega prometa, gasilske postaje in poštne poslovalnice, ki se nahajajo znotraj izbranega kraja.

Pri nekaterih možnostih za izbiro o oddaljenosti od objektov, se nahajajo tudi polja z besedilom, kamor lahko uporabnik vpiše naziv ali del naziva ustanove in išče glede na oddaljenost od točno določene ustanove, ali pa išče glede na vse izbrane ustanove v tem kraju.

Mesto in Tip nepremičnine sta obvezna parametra in morata biti obvezno izpolnjena pri vsakem iskanju.

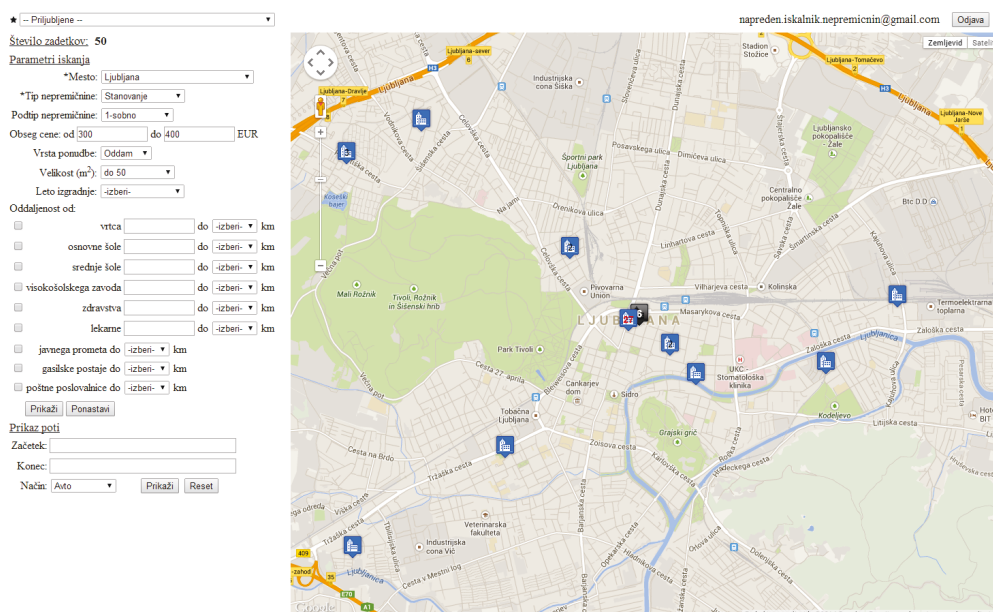
4.2.2 Prikaz rezultatov iskanja

Po izbiri parametrov in kliku na gumb Prikaži se center zemljevida prestavi na izbran kraj, na zemljevidu se pokažejo zaznamki z nepremičninami, zgoraj na levi strani iskalnik pa se s številko izpiše tudi število nepremičnin, ki ustrezajo iskanju. Če izbranim parametrom iskanja ne ustreza nobena nepremičnina, se uporabniku prikaže okno, ki ga na to opozori. Na Sliki 4.4 je prikazan primer uspešnega iskanja.

Zaznamki nepremičnin, ki se prikazujejo na zemljevidu so razdeljeni v dve skupini. Prva skupina z modrimi zaznamki predstavlja nepremičnine, ki vsebujejo tudi podatek o naslovu lokacije nepremičnine, druga skupina s črnimi zaznamki pa predstavlja nepremičnine brez točnega naslova. Črni zaznamki so prikazani na sredini mesta, in predstavljajo le informacijo o tem koliko nepremičnin je še v tem mestu. Če uporabnik zemljevid bolj približa, črni zaznamki izginejo iz zemljevida. Izgled obeh zaznamkov je prikazan na Sliki 4.5.

Za lažji pregled nad številom nepremičnin in ostalih objektov, ki se nahajajo na določenem območju se zaznamki istega tipa združujejo v en zaznamek, na katerem je izpisana številka, ki predstavlja koliko posameznih zaznamkov se nahaja na tem območju.

Ob kliku na gumb Ponastavi se vrednosti vseh parametrov iskanja ter prikaz zemljevida postavi na začetno vrednost.



Slika 4.4: Prikaz rezultatov ob uspešnem iskanju.



(a) Z naslovom

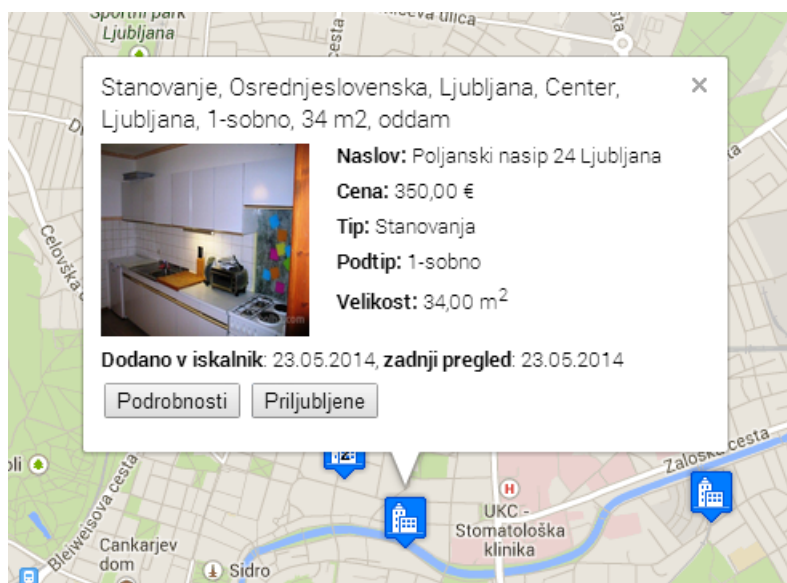


(b) Brez naslova

Slika 4.5: Zaznamka za prikaz nepremičnin.

4.2.3 Prikaz podatkov o nepremičnini

Klik na zaznamek nepremičnine modre barve uporabniku prikaže okno prikazano na Sliki 4.6, in vpiše naslov lokacije, kjer se nepremičnina nahaja, v polje Začetek v predelu za prikaz poti. Okno vsebuje osnovne podatke o nepremičnini: naziv, sliko, podatek o naslovu, ceni, tipu, podtipu in velikosti nepremičnine. Vsebuje pa tudi podatek o tem, kdaj je bila nepremičnina dodana v iskalnik in kdaj je bil nazadnje pregledan status nepremičnine v iskalniku.

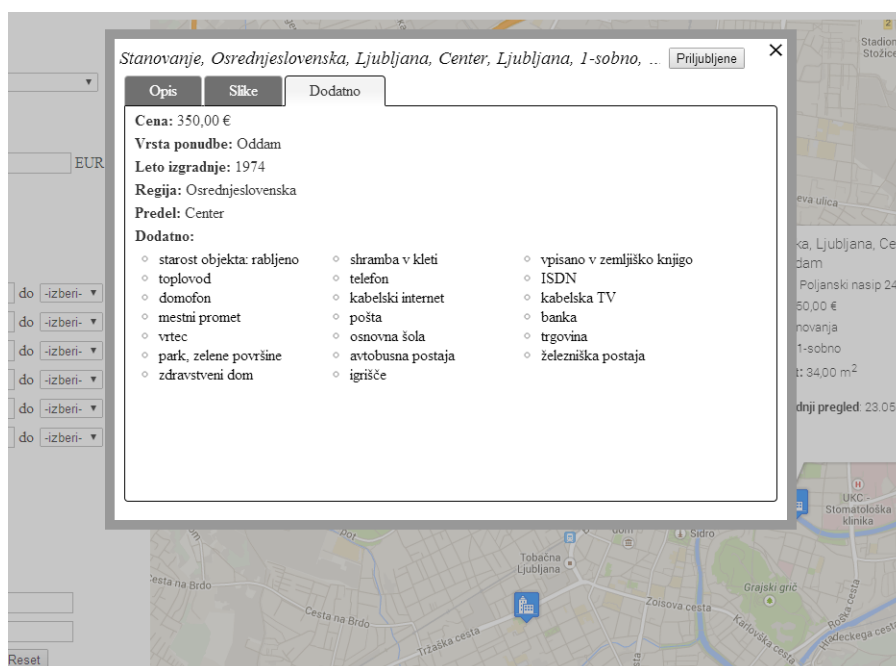


Slika 4.6: Okno za prikaz osnovnih podatkov o nepremičnini.

Klik na gumb Podrobnosti odpre novo okno, vidno na Sliki 4.7, kjer so prikazani dodatni podatki o nepremičnini. Okno vsebuje naziv nepremičnine in tri zavihke:

- **Opis** - vsebuje dodaten opis nepremičnine in gumb Podrobnosti. Klik na gumb Podrobnosti v novem zavihku odpre originalen oglas nepremičnine na strani bolha.com.
- **Slike** - uporabniku omogoča pregled slik nepremičnine.
- **Dodatno** - vsebuje podatke o ceni nepremičnine, vrsti ponudbe, leto izgradnje nepremičnine, regiji in predelu, kjer se nepremičnina nahaja ter dodatne lastnosti nepremičnine.

Obe okni, ki prikazujeta podatke o nepremičnini, vsebujeta tudi gumb Priljubljene. Klik na ta gumb izbrano nepremičnino doda med priljubljene za prijavljenega uporabnika, in gumb Priljubljene zamenja z gumbom Odstrani. S klikom na gumb Odstrani se nepremičnina odstrani iz seznama priljubljenih nepremičnin tega uporabnika.



Slika 4.7: Okno za prikaz dodatnih podatkov o nepremičnini.

4.2.4 Priljubljene nepremičnine

Vsak uporabnik si lahko ob ogledu nepremičnin najljubše shrani med priljubljene. Vse priljubljene nepremičnine se ob prijavi v aplikacijo prikažejo na zemljevidu in ostanejo vidne ves čas uporabe aplikacije.

Nazivi nepremičnin so shranjeni tudi v spustni seznam¹, ki se nahaja v zgornjem levem kotu zaslonske maske aplikacije. Ob izbiri ene izmed priljubljenih nepremičnin v seznamu, se na zemljevidu prikaže okno z osnovnimi podatki nepremičnine.

Zaznamki, ki na zemljevidu predstavljajo priljubljene nepremičnine, so oranžne in rdeče barve, in njun prikaz je na Sliki 4.8. Rdeči zaznamek (4.8b) predstavlja nepremičnine, ki so bile zaradi poteka oglasa odstranjene iz iskalnika. Ob kliku na tak zaznamek se prikaže okno, ki uporabniku sporoči, da ta nepremičnina ne obstaja več v iskalniku in ga vpraša, če želi odstraniti

¹angl. dropdown list

nepremičnino iz seznama. Če uporabnik potrdi, se nepremičnina odstrani iz seznama priljubljenih.



(a)



(b)

Slika 4.8: Zaznamka za prikaz priljubljenih nepremičnin.

4.2.5 Prikaz objektov

Poleg tega, da uporabnik lahko išče nepremičnine glede na oddaljenost od določenega objekta, si lahko lokacijo objektov tudi prikaže na zemljevidu. Objekte se lahko prikaže v okviru izvedbe iskanja nepremičnin in sicer tako, da se obkljuka polje pred imenom objekta. Vsak objekt je na zemljevidu prikazan s svojim zaznamkom, ki so predstavljeni na Sliki 4.9.

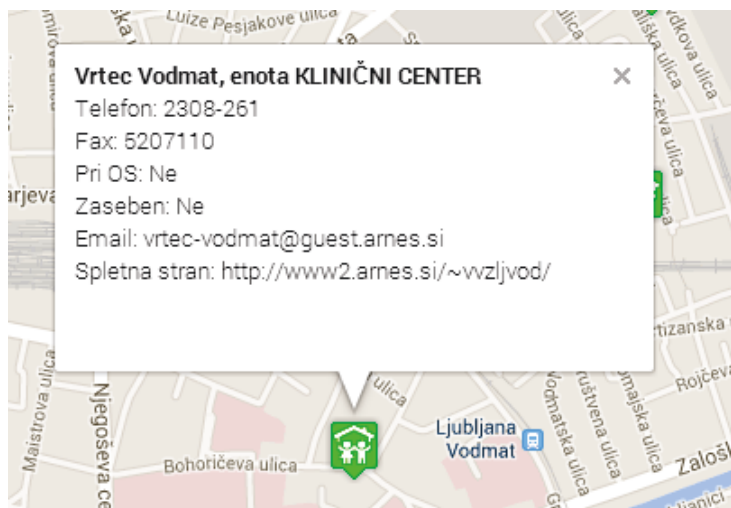
Ob kliku na zaznamek objekta se prikaže okno, ki vsebuje osnovne podatke o objektu. Podatki, ki so prikazani na oknu, so odvisni od tipa objekta. Na Sliki 4.10 je prikazano okno za prikaz podatkov o vrtcu.

4.2.6 Prikaz poti

Še ena funkcionalnost, ki jo aplikacija omogoča, je prikaz poti od začetnega do končnega naslova, ki jih lahko uporabnik vpiše v polja Začetek in Konec v predelu za prikaz poti. Uporabnik lahko izbere tudi način prevoza, ki ga želi med podanima lokacijama. Izbira lahko med štirimi načini potovanja: z lastnim avtomobilom, kolesom, javnim prevozom ali peš. Po kliku na gumb Prikaži se pot izriše kot krivulja na zemljevidu po cestah med lokacijo začetne in končne točke. V aplikaciji spodaj levo se izpiše tudi besedni opis poti, vsebuje pa tudi podatek o času, ki je potreben za potovanje z izbranim načinom prevoza. Primer prikaza poti od izbrane nepremičnine do Fakultete za računalništvo in informatiko je prikazan na Sliki 4.11.

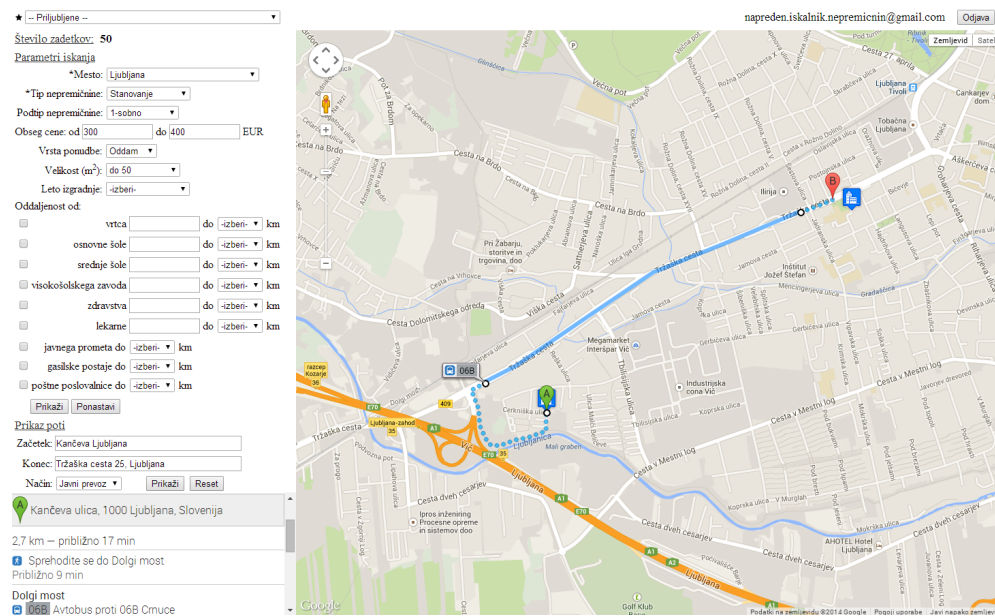


Slika 4.9: Zaznamki za prikaz dodatnih objektov.



Slika 4.10: Okno za prikaz podatkov o objektih.

Klik na gumb Reset postavi izbrane parametre na začetno vredno in zbriše besedni opis poti ter prikaz poti iz zemljevida.



Slika 4.11: Prikaz poti med dvema lokacijama.

Poglavje 5

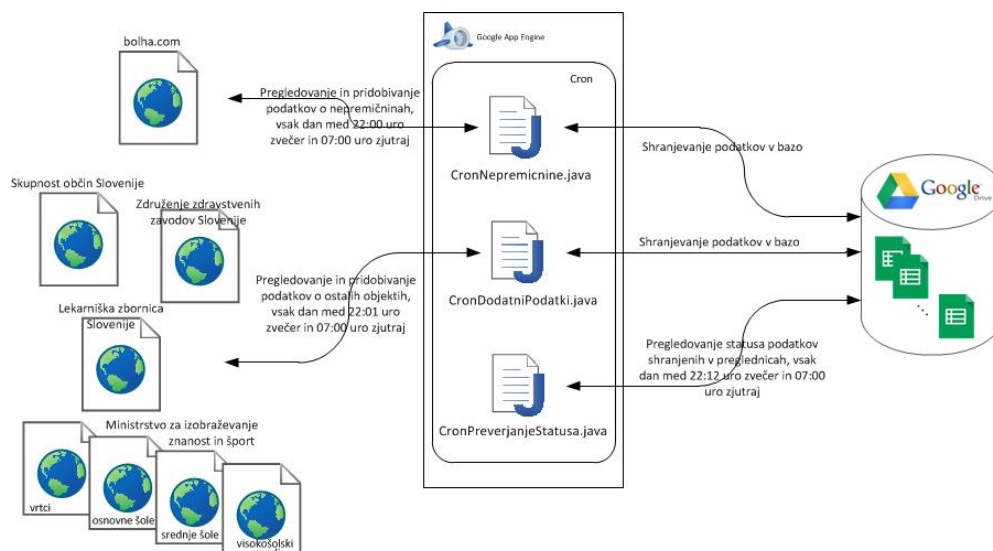
Implementacija iskalnika nepremičnin

5.1 Pridobivanje podatkov iz spleta

Izdelana aplikacija je zasnovana tako, da za uspešno delovanje potrebuje podatke iz drugi spletnih virov. Podatki o nepremičninah so pridobljeni s spletne strani bolha.com [1], podatki o občinah v Sloveniji s spletne strani Skupnosti občin Slovenije [39], podatki o vrtcih [40], osnovnih šolah [41], srednjih šolah [42] in visokošolski zavodih [43] s spletne strani Ministrstva za izobraževanje, znanost in šport, podatki o zdravstvenih ustanovah [44] s spletne strani Združenja zdravstvenih zavodov Slovenije in podatke o lekarnah [45] s spletne strani Lekarniške zbornice Slovenije.

Podatke iz omenjenih spletnih virov pa je za učinkovito delovanje aplikacije potrebno pridobiti samodejno. Na Sliki 5.1 je prikazan diagram, ki prikazuje komponente sistema, ki skrbijo za avtomatsko pridobivanje podatkov in shranjevanje teh podatkov v bazo ter pregledovanje statusa podatkov in brisanje zastarelih podatkov v bazi. Podrobnejši opis teh komponent bo sledil v nadaljevanju.

Za izvedbo samodejnega pridobivanja spletnih podatkov sem uporabila knjižnico HtmlUnit. Za to knjižnico sem se odločila, ker temelji na pro-



Slika 5.1: Diagram komponent sistema za avtomatsko pridobivanje podatkov.

gramskem jeziku Java ter omogoča dostop tudi do HTTPS strani in podpira strani z izvajanjem JavaScripta. Koda, ki uporablja knjižnico HtmlUnit je prikazana na Primeru 5.1 in predstavlja pridobitev podatkov o lekarnah v Sloveniji.

Primer 5.1: Koda za pridobitev podatkov o lekarnah.

```
WebClient webClient = new WebClient();
webClient.getOptions().setJavaScriptEnabled(true);
webClient.getOptions().setThrowExceptionOnScriptError(true);

HtmlPage pageLekarne = webClient.getPage("http://www.lzs.si/Mrezalekarn/
    Iskalnik/tabid/80/language/sl-SI/Default.aspx");
HtmlForm form = pageLekarne.getFormByName("Form");
HtmlSubmitInput button = form.getInputByName("dnn$ctr508$Seznam$cmdOK");
HtmlSelect select = (HtmlSelect) pageLekarne.getElementById("
    dnn_ctr508_Seznam_kr");
List<HtmlElement> options = select.getElementsByTagName("option");

for (int i = 1; i < options.size(); i++) {
    HtmlOption option = (HtmlOption) options.get(i);
    select.setSelectedAttribute(option, true);
    pageLekarne = button.click();
}
```

```

webClient.waitForBackgroundJavaScript(10 * 1000);
HtmlTable spletnaTabelaLekarne = (HtmlTable) pageLekarne.getElementById(
    "dnn_ctr508_Seznamlst");
if (spletnaTabelaLekarne != null) {
    for (int j = 3; j < spletnaTabelaLekarne.getRows().size(); j++) {
        HtmlTableRow vrstica = spletnaTabelaLekarne.getRows().get(j);
        String nazivLekarne = vrstica.getCells().get(1).asText();
        String nasl = vrstica.getCells().get(2).asText().split("\n")[0];
        String postnaSt = vrstica.getCells().get(2).asText().split("\n")
            [1].replaceAll("^\\s+|\\s+$", "").substring(0, 4);
        String posta = naslovPosta[1].substring(5);
        String[] drugo = vrstica.getCells().get(3).asText().split("\n");
        String telefon = "", faks = "", email = "", spletnaStran = "";
        for (int k = 0; k < drugo.length; k++) {
            if (drugo[k].contains("T:")) {
                telefon = drugo[k].replace("T:", "");
            }
            else if (drugo[k].contains("F:")) {
                faks = drugo[k].replace("F:", "");
            }
            else if (drugo[k].contains("E:")) {
                email = drugo[k].replace("E:", "");
            }
            else if (drugo[k].contains("I:")) {
                spletnaStran = drugo[k].replace("I:", "");
            }
        }
    }
}
}
}

```

Za vsako na novo pridobljeno nepremičnino oziroma drug objekt je potrebno pridobiti tudi točno lokacijo na Zemlji. Z uporabo Google Geocoding vmesnika se na podlagi podatka o naslovu objekta pridobi lokacijo v obliki podatka o pasovni širini in pasovni dolžini, ki se ga poda kot parameter HTTP zahtevi. Primer poizvedbe je prikazan na Primeru 5.2.

Primer 5.2: HTTP zahteva za pridobitev geografskih koordinat naslova.

```

http://maps.google.com/maps/api/geocode/json?sensor=false&address=Trzaska+
cesta+25,+Ljubljana

```

Zahteva vrne rezultate v JSON formatu. Najpomembnejši del rezultata je prikazan na Primeru 5.3.

Primer 5.3: Del rezultata HTTP zahteve na Primeru 5.2.

```

{
  "results": [
    {
      ...
      "geometry": {
        "location": {
          "lng": 14.4892307,
          "lat": 46.0448994
        }
        ...
      }
    }
  ],
  "status": "OK"
}

```

V aplikaciji za pridobivanje podatkov iz spleta in shranjevanje v preglednice skrbijo trije razredi:

- **CronNepremicnine.java** pridobiva podatke o nepremičninah na trgu,
- **CronDodatniPodatki.java** pridobiva podatke o krajih, vrtcih, osnovnih šolah, srednjih šolah, višjih strokovnih šolah, zdravstvenih ustanovah ter lekarnah in
- **CronPreverjanjeStatusa.java**, ki skrbi, da zastareli podatki ne ostanejo v bazi.

Problem pri izvajanju prej omenjenih razredov se je pojavil, ko je bilo potrebno zagotoviti, da se pridobivanje in osveževanje podatkov izvaja samodejno. Programi, ki to izvajajo v izdelani aplikaciji, se izvajajo dolgo časa in je potrebno, da se izvajajo redno. Za avtomatsko proženje načrtovanih nalog¹, ki se lahko začnejo izvajati ob točno določenem času (na primer vsak dan ob polnoči) ali v rednih časovnih intervalih (na primer vsakih deset minut) App Engine ponuja posebno storitev Cron [46], ki je v izdelani aplikaciji uporabljena za proženje programov, ki upravljajo s podatki. Vendar pa se vsaka sprožena naloga lahko izvaja le največ 10 minut. Problem omejenega

¹angl. scheduled tasks

izvajanja je bil delno rešen s tem, da si vsak program v svojo tabelo, ki se nahaja v preglednici Baza, zapisuje, kaj je nazadnje naredil in ob naslednjem klicu nadaljuje od tam naprej. Še vedno pa obstaja problem, da po 10 minutah, Cron Service prekine izvajanje programa in si v dnevnik shrani napako.

V aplikaciji se, kot je bilo prej omenjeno uporabljajo tri cron storitve, katerih klici za izvajanje so prikazani na Primeru 5.4. Interval izvajanja je izbran tako, da se ob enakem času ne izvajata dva programa, ki dostopata do istih tabel v preglednicah.

Primer 5.4: Dokument cron.xml v aplikaciji.

```
<?xml version="1.0" encoding="UTF-8"?>
<cronentries>
  <cron>
    <url>/cron/cronjobnepremicnine</url>
    <description>Nadaljuj pregled nepremicnin vsakih 23 minut med 22:00 uro
      zvečer in 07:00 uro zjutraj</description>
    <schedule>every 23 minutes from 22:00 to 07:00</schedule>
    <timezone>Europe/Ljubljana</timezone>
  </cron>
  <cron>
    <url>/cron/cronjobdodatnipodatki</url>
    <description>Nadaljuj pregled dodatnih podatkov vsakih 23 minut med
      22:01 uro zvečer in 07:00 uro zjutraj</description>
    <schedule>every 23 minutes from 22:01 to 07:00</schedule>
    <timezone>Europe/Ljubljana</timezone>
  </cron>
  <cron>
    <url>/cron/cronjobpreverjanjestatusa</url>
    <description>Pregleda status podatkov vsakih 23 minut med med 22:12 uro
      zvečer in 07:00 uro zjutraj</description>
    <schedule>every 23 minutes from 22:12 to 07:00</schedule>
    <timezone>Europe/Ljubljana</timezone>
  </cron>
</cronentries>
```

5.2 Shranjevanje podatkov

Aplikacija za podatkovno bazo uporablja javne Google preglednice. V ta namen je bil kreiran nov Google račun napreden.iskalnik.nepremicnin@google.com,

ki ima shranjene te preglednice v Google Drive. Zaradi omejitve števila celic s podatki na 2 milijona celic aplikacija za shranjevanje podatkov uporablja 5 preglednic, tri za podatke o nepremičninah, eno za podatke o priljubljenih nepremičninah uporabnikov in eno za preostale podatke:

- **Nepremicnine_podatki** vsebuje delovni list **Nepremicnine**, ki shranjuje osnovne podatke o nepremičnini: tip, podtip, vrsta ponudbe, url, naziv, cena, regija, kraj, predel, velikost, leto izgradnje, naslov, opis, pasovna širina in dolžina lokacije nepremičnine, spletni naslov naslovne slike nepremičnine.
- **Nepremicnine_dodatne_lastnosti** vsebuje delovni list **Dodatne_lastnosti**, ki shranjuje podatke o dodatnih lastnostih nepremičnine.
- **Nepremicnine_slike** vsebuje delovni list **Slike**, ki shranjuje podatke o spletnem naslovu posameznih slik nepremičnine.
- **Priljubljene_nepremicnine** vsebuje delovni list **Priljubljene**, ki shranjuje identifikator nepremičnine ter pasovno širino in dolžino lokacije nepremičnine, ki si jih uporabniki izberejo za priljubljene.
- **Baza** shranjuje delovne liste za vse ostale podatke uporabljene v aplikaciji:
 - ◊ **Kraji** - osnovni podatki ter podatek o pasovni širini in dolžini lokacije za vsako občino v Sloveniji.
 - ◊ **Vrtci** - osnovni podatki o vrtcih in njihovih enotah ter tudi podatek o tem, ali se vrtec nahaja pri osnovni šoli in če je zaseben ali javen.
 - ◊ **Osnovne_sole** - osnovni podatki o osnovnih šolah.
 - ◊ **Srednje_sole** - osnovni podatki o srednjih šolah.
 - ◊ **Visje_strokovne_sole** - osnovni podatki o visokošolskih zavodih.

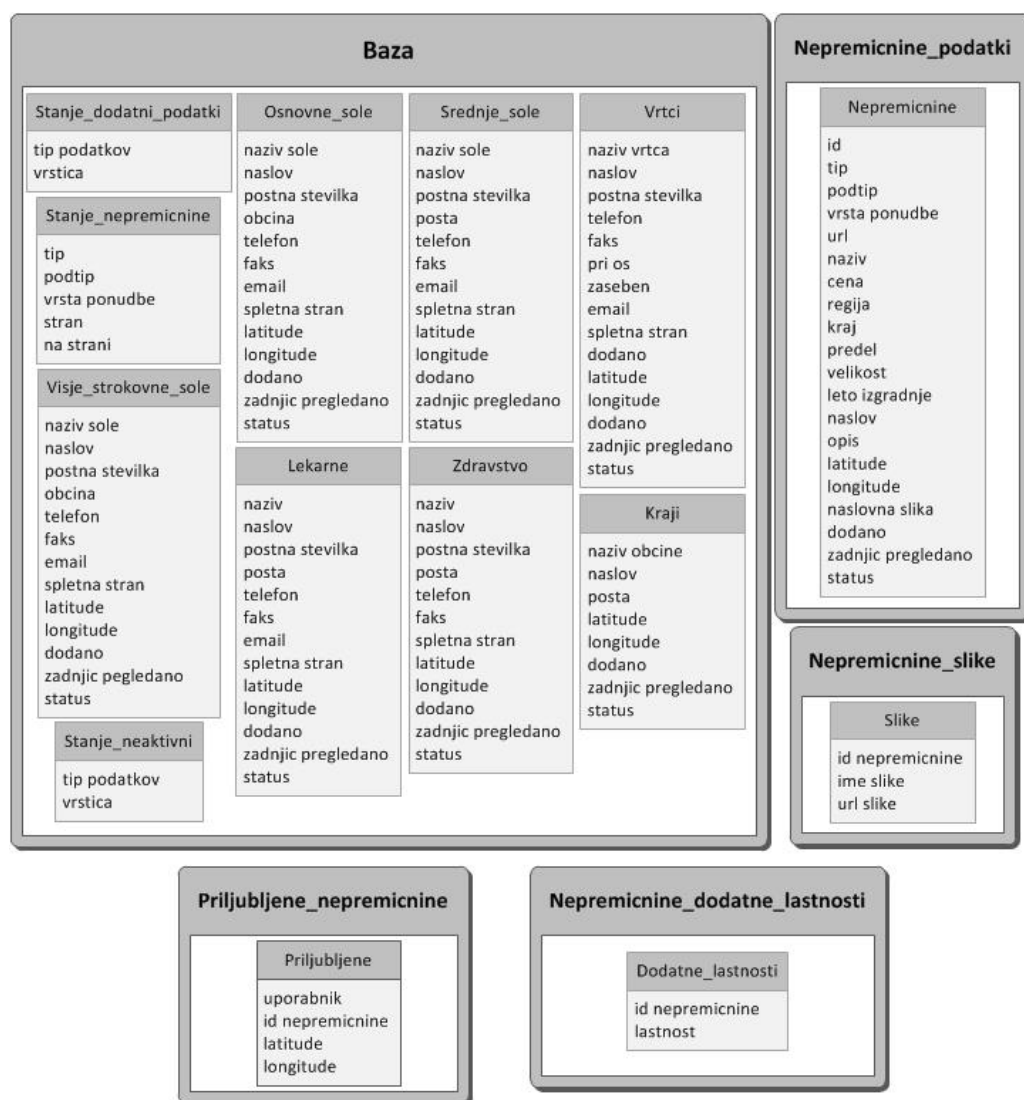
- ◇ **Zdravstvo** - osnovni podatki o zdravstvenih zavodih.
- ◇ **Lekarne** - osnovni podatki o lekarnah.
- ◇ **Stanje_nepremicnine** - podatki o tem, kje je končal program za pregledovanje nepremičnin. V eni vrstici se hrani podatek o tipu nepremičnine, podtipu nepremičnine, vrsti ponudbe, številki strani in številki nepremičnine na strani.
- ◇ **Stanje_dodatni_podatki** - podatki o tem, kje je končal program za pregledovanje podatkov o objektih. V eni vrstici se hrani podatek o tipu podatkov in številki vrstice.
- ◇ **Stanje_neaktivni** - podatki o tem, kje je končal program pri pregledovanju statusov shranjenih podatkov v tabelah. V eni vrstici se hrani podatek o tipu podatkov in številki vrstice.

Tabele v preglednicah `Nepremicnine_podatki` in `tabele_s_podatki` o objektih v preglednici Baza poleg omenjenih podatkov vsebujejo tudi podatke o tem, kdaj je bil podatek dodan v tabelo, kdaj je bilo stanje podatke nazadnje pregledano in kakšen je status podatka. Grafični prikaz preglednic je viden na Sliki 5.2.

Za shranjevanje in pridobivanje podatkov iz preglednic se v programih uporablja vmesnik Google Spreadsheet API verzija 3.0 [47]. Vmesnik omogoča funkcionalnosti, kot so pridobivanje seznama in podatkov o preglednicah, kreiranje in brisanje delovnih listov, dodajanje, spreminjanje in brisanje vrstic v tabelah, pridobivanje podatkov shranjenih v vrsticah tabele.

Za uporabo vmesnika v Java programu je potrebno najprej izvesti inicializacijo in odobriti dostop za določenega uporabnika. V tem primeru gre za dostop aplikacije do preglednic, zato je uporabljen OAuth 2.0 za interakcijo strežnik-strežnik. Ta način namesto računa posameznega končnega uporabnika potrebuje račun storitve, ki pripada aplikaciji. In s pomočjo tega računa in pripadajočega zasebnega ključa² aplikacija podpiše zahtevo oziroma žeton

²angl. private key



Slika 5.2: Preglednice s tabelami in njihovimi stolpci.

za dostop. Primer 5.5 prikazuje programsko kodo potrebno za dostop do preglednic v Google Drive, ki je uporabljena v aplikaciji.

Primer 5.5: Programska koda za dostop do Google preglednic z OAuth 2.0.

```
String CLIENT_ID = "711446868423-nfjmhs7ftcjb2okck27o8t18765ofa3v.apps.\n    googleusercontent.com";  
String CLIENT_SECRET = "BDq7BF5r51yhHJLUp8bHoRFe";  
final String SERVICE_ACCOUNT_EMAIL = "711446868423-\n    nfjmhs7ftcjb2okck27o8t18765ofa3v@developer.gserviceaccount.com";  
final String SERVICE_ACCOUNT_PKCS12_FILE_PATH = "key.p12";  
List<String> scopes = new ArrayList<String>();  
scopes.add("https://spreadsheets.google.com/feeds");  
  
HttpTransport HTTP_TRANSPORT = new NetHttpTransport();  
JsonFactory JSON_FACTORY = new JacksonFactory();  
  
GoogleCredential credential = new GoogleCredential.Builder()  
    .setTransport(HTTP_TRANSPORT)  
    .setJsonFactory(JSON_FACTORY)  
    .setServiceAccountId(SERVICE_ACCOUNT_EMAIL)  
    .setServiceAccountScopes(scopes)  
    .setServiceAccountPrivateKeyFromP12File(new File(  
        SERVICE_ACCOUNT_PKCS12_FILE_PATH))  
    .build();  
SpreadsheetService service = new SpreadsheetService("com.cron.jobs");  
service.setOAuth2Credentials(credential);  
  
URL spreadsheetUrl = new URL("https://spreadsheets.google.com/feeds/\n    spreadsheets/private/full");
```

Ko je dostop do preglednic zagotovljen, se lahko izvaja vse ostale funkcionalnosti, ki jih ponuja vmesnik.

5.3 Uporabniški vmesnik

V prejšnjem poglavju je bilo delovanje iskalnika opisano s strani uporabnika, v nadaljevanju pa bodo vse funkcionalnosti predstavljene še s tehničnega vidika.

nahaja v datoteki `mainpage.jsp`. Ob prihodu na glavno stran se najprej nastavi parametre za zemljevid. Aplikacija uporablja Google zemljevid cest in v osnovi je center zemljevida postavljen na sredino Slovenije. Inicializacija Google zemljevida je prikazana na Primeru 5.7.

Primer 5.7: Programska koda za inicializacijo Google zemljevida.

```
var globalneSpremenljivke = {
    map: ""
};

google.maps.event.addDomListener(window, 'load', initialize);

function initialize() {
    var mapOptions = {
        zoom: 8,
        center: new google.maps.LatLng(46.151241, 14.995463),
        mapTypeId: google.maps.MapTypeId.ROADMAP
    };

    globalneSpremenljivke.map = new google.maps.Map(document.getElementById(
        "map-canvas"), mapOptions);
}
```

Ko uporabnik izbere parametre iskanja in začne iskanje, aplikacija pridobi podatke iz Google preglednice. Vsaka preglednica ter tudi delovni list v Google Drive ima svoj enolični identifikator in če je preglednica objavljena na spletu, ima določen spletni naslov, preko katerega lahko vidimo podatke v preglednici, brez da bi se bilo potrebno prijaviti. Vse preglednice s podatki, ki se uporabljajo v aplikaciji, so objavljene na spletu, zato se za pridobitev podatkov uporablja metoda `jQuery.getJSON(url, [data], [callback])`, ki z uporabo AJAX HTTP GET zahteve vrne podatke v JSON obliki. Metodi se poda parameter `url`, ki predstavlja strukturirano poizvedbo in parameter `[callback]`, ki predstavlja funkcijo, ki se izvede ob uspešni izvedbi zahteve.

Podatki o lokacijah in drugih informacijah za avtobusne in železniške postaje javnega prometa, poštne poslovalnice ter gasilske postaje niso shranjeni v tabelah, ampak se jih za vsako iskanje pridobi s pomočjo knjižnice Places. Za iskanje avtobusnih in železniških postaj se uporablja tip iskanja `nearby search` (bližnje iskanje), ki išče glede na tip lokacije, *bus_station* za avtobusne

postaje in *train_station* za železniške postaje in išče tiste lokacije, ki se nahajajo znotraj 10 kilometrskega radija, kjer center kroga predstavlja središče izbranega kraja. Problem tega iskanja je, da lahko išče le med podatki, ki jih ima na voljo Google, in zelo malo je podatkov o avtobusnih postajah v Sloveniji. Na voljo so le podatki o nekaterih postajah na območju Ljubljane. Za iskanje poštних poslovalnic in gasilskih postaj pa se uporablja text search (iskanje s ključnimi besedami) in išče tiste lokacije, ki se nahajajo znotraj 5 kilometrskega radija, kjer center kroga prav tako predstavlja središče izbranega kraja. Iskalni niz za iskanje poštних poslovalnic je »postna poslovalnica«, za iskanje gasilskih postaj pa je »gasilsko društvo«.

Vsaka nepremičnina in ostali objekti so na zemljevidu prikazani z Google zaznamkom [49]. Zaznamek za prikaz na zemljevidu obvezno potrebuje podatek o pasovni širini in dolžini lokacije ter podatek o zemljevidu na katerem naj se prikaže. Zaznamek za svoj prikaz lahko uporablja privzeto ali kakršnokoli drugo sliko. Slike, ki so uporabljene za prikaz zaznamka, se imenujejo tudi ikone. V tej aplikaciji so uporabljene ikone iz projekta Map Icons Collection [50], kjer je na voljo več ko 700 različnih brezplačnih ikon. Vsaka od teh ikon je izdelana v sedmih različnih stilih in je lahko poljubne barve. Razvijalec si lahko tako sam izbere stil in barvo za izbrano sliko. Ikone, ki se iz tega projekta uporabljajo v aplikaciji, so prikazane na Slikah 4.5, 4.8, 4.9. Ker se na zemljevidu lahko pojavi veliko število zaznamkov istega tipa, je zaradi lepše preglednosti implementirano tudi grupiranje zaznamkov posameznih tipov z uporabo knjižnice MarkerClusterer [51]. Grupiranje deluje tako, da pregleduje zaznamke znotraj skupine zaznamkov in vsakega doda k najbližji grupi³, znotraj meja katere se nahaja.

Vsakemu zaznamku posebej se lahko določi tudi, kaj se zgodi ob kliku nanj. V naši aplikaciji se ob kliku na posamezen zaznamek prikaže okno tipa InfoWindow [52], na katerem so prikazani podatki. Uporabljata se dva različna okna, eden za prikaz osnovnih podatkov o nepremičnini, drugi pa za prikaz podatkov o ustanovah. Naenkrat sta lahko prikazani dve okni, od

³angl. cluster

vsakega tipa eden. Obe okni sta definirani ob prihodu na glavno stran v funkciji `initialize()`.

Na Primeru 5.8 je prikazana programska koda, ki predstavlja implementacijo pridobitve podatkov o vrtcih iz preglednice ter kreiranje, prikaz in grupiranje zaznamkov na zemljevidu.

Primer 5.8: Programska koda za pridobitev in prikaz vrtcev na zemljevidu.

```
var markersVrtci = [];
var infowindowData = new google.maps.InfoWindow();
$.getJSON('https://spreadsheets.google.com/feeds/list/1
JFsl4M76xcjXWaMPnox8oW3dzdB10.6T_4nO2vcVqTE/707867289/public/values?alt=
json&sq=obcina="Ljubljana"', function(json) {
    if (!jQuery.isEmptyObject(json.feed.entry)) {
        for (var i = 0; i < json.feed.entry.length; i++) {
            var latlng = new google.maps.LatLng(json.feed.entry[i].
                gsx$latitude.$t, json.feed.entry[i].gsx$longitude.$t);
            var marker = new google.maps.Marker({
                position: latlng,
                map: globalneSpremenljivke.map,
                icon: 'map_icons/daycare.png'
            });

            var content = '<div id="infoClosest" style="width:290px; height
:165px">' + '<div id="infoNaziv"><b>' + json.feed.entry[i].
                gsx$nazivvrtca.$t + '</b></div>' + '<div id="infoNaslov">' +
                json.feed.entry[i].gsx$naslov.$t + ', ' + json.feed.entry[i]
                .gsx$postnastevilka.$t + ' ' + json.feed.entry[i].
                gsx$obcina.$t + '</div>' + '<div id="infoTelefon">Telefon: '
                + json.feed.entry[i].gsx$telefon.$t + '</div>' + '<div id="
                infoFax">Fax: ' + json.feed.entry[i].gsx$faks.$t + '</div>'
                + '<div id="infoPriOS">Pri OS: ' + json.feed.entry[i].
                gsx$prios.$t + '</div>' + '<div id="infoZaseben">Zaseben: '
                + json.feed.entry[i].gsx$zaseben.$t + '</div>' + '<div id="
                infoEmail">Email: ' + json.feed.entry[i].gsx$email.$t + '</
                div>' + '<div id="infoSpletnaStran">Spletna stran: ' + json.
                feed.entry[i].gsx$spletnastran.$t + '</div>' + '</div>';

            google.maps.event.addListener(marker, 'click', (function(marker,
                content){
                    return function() {
                        infowindowData.setContent(content);
                        infowindowData.open(globalneSpremenljivke.map, marker);
                    };
                })(marker, content));
            markersVrtci.push(marker);
```

```

    }
    var mcOptions = {
      styles: [{
        textColor: 'black',
        textSize: 13,
        url: 'map_icons/daycare.png',
        height: 37,
        width: 32
      },
      {
        textColor: '#E62E00',
        textSize: 14,
        url: 'map_icons/daycare.png',
        height: 37,
        width: 32
      },
      {
        textColor: 'red',
        textSize: 14,
        url: 'map_icons/daycare.png',
        height: 37,
        width: 32
      }
    ],
    maxZoom: 18
  };
  var markerVrtecCluster = new MarkerClusterer(globalneSpremenljivke.
    map, markersVrtci, mcOptions);
}
});

```

Primer 5.9 pa predstavlja izsek iz JSON podatkov, pridobljenih v Primeru 5.8.

Primer 5.9: Podatki o vrtcu v JSON formatu.

```

{
  ...
  feed: {
    ...
    entry: [
      {
        gsx$nazivvrtca: {
          $t: "Vrtec Pod Gradom"
        },
        gsx$naslov: {
          $t: "Praprotnikova ulica 2"
        },

```



```
    gsx$postnastevilka: {  
      $t: "1000"  
    },  
    gsx$obcina: {  
      $t: "Ljubljana"  
    },  
    gsx$telefon: {  
      $t: "01 241 26 00"  
    },  
    gsx$faks: {  
      $t: "01 241 26 10"  
    },  
    gsx$prios: {  
      $t: "Ne"  
    },  
    gsx$zaseben: {  
      $t: "Ne"  
    },  
    gsx$email: {  
      $t: "vrtec-pod-gradom@guest.arnes.si"  
    },  
    gsx$spletnastran: {  
      $t: "www.vrtec-podgradom.org"  
    },  
    gsx$latitude: {  
      $t: "46.0432553"  
    },  
    gsx$longitude: {  
      $t: "14.510436"  
    },  
    gsx$dodano: {  
      $t: "05.05.2014"  
    },  
    gsx$zadnjicpregledano: {  
      $t: "23.05.2014"  
    },  
    gsx$status: {  
      $t: "aktiven"  
    }  
  },  
  ...  
]  
}
```

Prej omenjeno okno za prikaz osnovnih podatkov o nepremičninah vsebuje tudi gumb Podrobnosti, ob kliku na katerega se prikaže novo okno, ki

vsebuje podrobnejše podatke o nepremičnini. Za implementacijo funkcionalnosti okna ni bila uporabljena nobena posebna knjižnica, ampak le HTML elementi in JavaScript funkcije. Pomembno vlogo za prikaz okna ima datoteka `overlayWindow.css`, ki vsebuje definicije stilov HTML elementov, ki se nahajajo na tem oknu.

Stili ostalih HTML elementov, ki se pojavljajo v aplikaciji, se nahajajo v datotekah `googleMap.css`, `infoBox.css` in `resultsPage.css`.

5.3.3 Shranjevanje priljubljenih nepremičnin

Ko uporabnik želi shraniti izbrano nepremičnino med priljubljene s klikom na gumb Priljubljene na enem izmed oken, se najprej gumb Priljubljene zamenja z gumbom Odstrani. Spremeni se tudi ikono zaznamka s klicem funkcije `setIcon()`. Naziv izbrane nepremičnine se doda med opcije v spustnem seznamu priljubljenih, identifikator nepremičnine, ime uporabnika ter tudi pasovna širina in dolžina lokacije nepremičnine pa se doda v preglednico z AJAX klicem funkcije servlet razreda `NapredenIskalnikNepremicninServlet.java`, ki je prikazan na Primeru 5.10. Podatek o pasovni širini in dolžini lokacije nepremičnine se shrani zato, da se v primeru, če je nepremičnina že odstranjena iz iskalnika lahko še vseeno postavi zaznamek na zemljevid. Funkcija v servlet razredu za shranjevanje pridobljenih podatkov AJAX klika v preglednice uporablja Google Spreadsheet vmesnik, ki je omenjen na začetku tega poglavja.

Primer 5.10: AJAX klic za shranjevanje priljubljenih nepremičnin.

```
jQuery.ajax({
    url: "napredeniskalniknepremicnin",
    data: "action=dodajPriljubljene&imeUporabnika=uporabniko_ime_uporabnika
        &idPriljubljene=id_nepremicnine&latitude=pasovna_sirina_nepremicnine
        &longitude=pasovna_dolzina_nepremicnine"
    cache: false,
    success: function(data) {
    }
});
```

Podobno kot dodajanje nepremičnine med priljubljene, je izdelana tudi funkcionalnost odstranjevanja nepremičnine iz priljubljenih. Ob kliku na gumb Odstrani se najprej na obeh oknih gumb Odstrani zamenja z gumbom Priljubljene. Naziv nepremičnine se odstrani iz opcij v spustnem seznamu priljubljenih, nepremičnina pa se odstrani tudi iz preglednice priljubljenih nepremičnin. AJAX klic je enak kot na Primeru 5.10, razlika je le v vrednosti parametra data, ki je tu "action=odstraniPriljubljene&imeUporabnika=uporabniksko_ime_uporabnika&idPriljubljene=id_nepremicnine".

5.3.4 Prikaz poti

Za prikaz poti od začetne do končne lokacije se v aplikaciji uporablja Directions service. Storitev omogoča prikaz poti s štirimi različnimi načini potovanja:

- **lasten avtomobil** (google.maps.TravelMode.DRIVING): je privzeti način potovanja in za prikaz poti uporabi cestno omrežje,
- **kolo** (google.maps.TravelMode.BICYCLING): za prikaz poti uporabi kolesarske poti in prednostne⁴ ceste,
- **javni prevoz** (google.maps.TravelMode.TRANSIT): za prikaz poti uporabi linije javnih prevoznih sredstev in
- **peš** (google.maps.TravelMode.WALKING): za prikaz poti uporabi pešpoti in poti s pločniki.

Tudi v iskalniku se lahko prikaže pot s katerikoli izmed zgoraj naštetim tipom potovanj, vendar se pojavlja problem pri tipih javni prevoz in kolo, saj Google za Slovenijo nima podatkov o vseh linijah javnih prevozov in kolesarskih poteh. Zato se v iskalniku velikokrat za ta dva načina potovanja pot ne izpiše. Koda, ki se uporablja za prikaz poti, je predstavljena v Primeru 5.11.

⁴angl. preferred

Primer 5.11: Programska koda za prikaz poti med točkama.

```
var directionsDisplay = new google.maps.DirectionsRenderer();
var directionsService = new google.maps.DirectionsService();

directionsDisplay.setMap(map);
directionsDisplay.setPanel(document.getElementById("directions-panel"));

function calcRoute() {
    var start = document.getElementById("start").value;
    var end = document.getElementById("end").value;

    var selectedMode = document.getElementById("mode").value;
    var request = {
        origin: start,
        destination: end,
        travelMode: google.maps.TravelMode[selectedMode]
    };

    globalneSpremenljivke.directionsService.route(request, function(response
    , status) {
        if (status == google.maps.DirectionsStatus.OK) {
            globalneSpremenljivke.directionsDisplay.setDirections(response);
        }
    });
}
```

Poglavje 6

Zaključek

V diplomski nalogi je bila razvita spletna aplikacija za iskanje nepremičnin v Sloveniji, ki združuje podatke iz različnih spletnih virov in s tem uporabniku olajša iskanje. V aplikacijo so poleg podatkov o nepremičninah vključeni tudi podatki o vrtcih, osnovnih šolah, srednjih šolah, visokošolskih zavodih, zdravstvenih zavodih in lekarnah, ki so pridobljeni iz različnih spletnih virov, ter tudi podatki o železniških in avtobusnih postajah, poštnih poslovalnicah in gasilskih postajah, ki so pridobljeni s pomočjo Google iskanja. Izbira o tem, kateri podatki so vključeni v aplikacijo, je temeljila na pomembnosti posameznih podatkov, ki vplivajo na izbiro nepremičnine in dosegljivostjo teh podatkov na spletu. Aplikacija torej poleg iskanja po osnovnih kriterijih nepremičnin omogoča tudi iskanje nepremičnin glede na oddaljenost od prej omenjenih objektov. Vsebuje tudi druge funkcionalnosti kot so prikaz nepremičnin in drugih objektov na Google zemljevidu, prikaz poti med dvema lokacijama kot krivulja na zemljevidu ter kot besedni opis, prijava v aplikacijo z Google računom in možnost izbire priljubljenih nepremičnin in prikaz osnovnih podatkov o nepremičninah in ostalih objektih.

S tehničnega vidika je aplikacija napisana v programskih jezikih Java in JavaScript in se izvaja na strežniku Google App Engine. Za podatkovno bazo so uporabljene preglednice, ki so shranjene v Google Drive. Podatki o nepremičninah in ostalih objektih, ki so iz spletnih virov pridobljeni s

pomočjo tehnike luščenja spletnih podatkov in knjižnice HtmlUnit. Celotna izvorna koda aplikacije se nahaja na Bitbucketu [53].

Med razvojem aplikacije se je pojavilo tudi nekaj težav in ovir. Prva težava je ta, da oglasi z nepremičninami zelo redko vsebujejo točen naslov, kje se nepremičnina nahaja, zato aplikacija težko prikaže realno sliko na zemljevidu, pa tudi veliko nepremičnin je lahko zaradi tega prikazanih na istem mestu in uporabnik lahko vidi podrobne podatke le o eni nepremičnini. Tudi naslednja težava se nanaša na nepopolnosti v podatkih, saj na spletu ni mogoče najti točnih lokacij vseh avtobusnih postaj v Sloveniji. Zato so v aplikaciji uporabljeni podatki, ki jih ima na voljo Google, vendar pa je tudi tu na voljo zelo malo postaj, kar vpliva na to, da iskanje po oddaljenosti od javnega prometa ni zanesljivo oziroma ne vrne veliko rezultatov. Zadnjo izmed težav predstavlja občasno počasna odzivnost preglednic v Google Drive, ko je iz njih potrebno pridobiti podatke, ki se uporabljajo v iskalniku.

Izdelan iskalnik nepremičnin združuje kar nekaj več funkcionalnosti kot jih ponujajo podobni iskalniki v slovenskem prostoru, vendar pa ima tudi ta iskalnik možnosti za izboljšavo. Med prvimi izboljšavami bi lahko bila podatkovna baza, ki bi imela boljšo odzivnost in bi omogočala hitrejši dostop. V uporabniški vmesnik bi se lahko za tiste nepremičnine, kateri oglasi ne vsebujejo naslova, dodal seznam, ki bi za vsako nepremičnino vseboval sliko in nekaj osnovnih podatkov. Dobro bi bilo tudi, če bi se pridobilo lokacije vseh postaj javnega prometa v Sloveniji. Prav tako pa bi se lahko izboljšal tudi sam videz uporabniškega vmesnika, ki bi za uporabnike še izboljšal izkušnjo iskanja, saj na njem v tej diplomski nalogi ni bilo velikega poudarka.

Literatura

- [1] Iskalnik bolha.com.
<http://www.bolha.com/nepremicnine>. Pridobljeno 20.1.2014.
- [2] Iskalnik Nepremičnine.net:
<http://www.nepremicnine.net>. Pridobljeno 18.5.2014.
- [3] Iskalnik GoHome.si:
<http://www.gohome.si>. Pridobljeno 5.6.2014.
- [4] Iskalnik salomon.si:
<http://www.salomon.si>. Pridobljeno 5.6.2014.
- [5] Iskalnik Nepremičninar.com:
<http://www.nepremicninar.com>. Pridobljeno 5.6.2014.
- [6] Javni vpogled v nepremičnine na Geodetski upravi Republike Slovenije. Dostopno na:
<http://prostor3.gov.si/javni>. Pridobljeno 5.6.2014.
- [7] Iskalnik Domart:
<http://www.domart.si>. Pridobljeno 5.6.2014.
- [8] Iskalnik Hiša nepremičnine:
<http://www.hisanepremicnine.si>. Pridobljeno 5.6.2014.
- [9] A. Henry. AutNo Cuts Down Your Commute, Finds You An Apartment Close to Mass Transit, 2013. Dostopno na:

- <http://lifehacker.com/5992701/autno-cuts-down-your-commute-finds-you-an-apartment-close-to-mass-transit>. Pridobljeno 8.1.2014.
- [10] Spletna stran iskalnika nepremičnin AutNo:
<http://www.autno.com>. Pridobljeno 8.1.2014.
- [11] Lestvica petnajstih najbolj popularnih iskalnikov nepremičnin.
Dostopno na: <http://www.ebizmba.com/articles/real-estate-websites>.
Pridobljeno 18.5.2014.
- [12] Spletna stran iskalnika nepremičnin Zillow:
<http://www.zillow.com>. Pridobljeno 19.5.2014.
- [13] Spletna stran iskalnika nepremičnin Trulia:
<http://www.trulia.com>. Pridobljeno 19.5.2014.
- [14] Eclipse. Dostopno na:
<http://www.eclipse.org>. Pridobljeno 10.1.2014.
- [15] Google razširitev za Eclipse. Dostopno na:
<https://developers.google.com/eclipse>. Pridobljeno 10.1.2014.
- [16] Google App Engine. Dostopno na:
<https://developers.google.com/appengine>. Pridobljeno 10.1.2014.
- [17] M. Malawski, M. Kuźniar, P. Wójcik, M. Bubak, "How to Use Google App Engine for Free Computing", *IEEE Internet Computing*, št. 17, zv. 1, str. 50–59, 2013.
- [18] A. Zahariev. "Google App Engine",
Current Internet Trends Seminar on Internetnetworking, 2009. Dostopno na:
http://www.cse.hut.fi/en/publications/B/5/papers/1Zahariev_final.pdf.
Pridobljeno 6.6.2014.
- [19] K. Roche, J. Douglas. Beginning JavaTM Google App Engine, 2009, poglavje 9.

-
- [20] Administracijska konzola. Dostopno na:
<https://console.developers.google.com>. Pridobljeno 10.1.2014.
- [21] Google Drive. Dostopno na:
<https://drive.google.com>. Pridobljeno 27.1.2014.
- [22] Google Maps Javascript API. Dokumentacija dostopna na:
<https://developers.google.com/maps/documentation/javascript>.
Pridobljeno 15.1.2014.
- [23] Storitev Geocoding. Dokumentacija dostopna na:
<https://developers.google.com/maps/documentation/javascript/geocoding>. Pridobljeno 15.1.2014.
- [24] Storitev Directions. Dokumentacija dostopna na:
<https://developers.google.com/maps/documentation/javascript/directions>. Pridobljeno 30.1.2014.
- [25] Knjižnica Places. Dokumentacija dostopna na:
<https://developers.google.com/maps/documentation/javascript/places>.
Pridobljeno 11.3.2014.
- [26] Dokumentacija objekta PlaceResult. Dostopno na:
<https://developers.google.com/maps/documentation/javascript/reference#PlaceResult>. Pridobljeno 11.3.2014.
- [27] Seznam podprtih tipov iskanja v knjižnici Places. Dostopno na:
https://developers.google.com/places/documentation/supported_types.
Pridobljeno 11.3.2014.
- [28] D. Hardt. The OAuth 2.0 Authorization Framework, RFC6749, 2012.
- [29] B. Leiba, "OAuth Web Authorization Protocol", *IEEE Internet Computing*, št. 16, zv. 1, str. 74–77, 2012.

-
- [30] Uporaba OAuth 2.0 za dostop do Google vmesnikov. Dostopno na: <https://developers.google.com/accounts/docs/OAuth2>. Pridobljeno 13.5.2014.
- [31] M. Turland. *php|architect's Guide to Web Scraping*, 2010.
- [32] J. Wagnon. *Web Scraping - Data Collection or Illegal Activity?*, 2013. Dostopno na: https://devcentral.f5.com/articles/web-scraping-data-collection-or-illegal-activity#.U5MU33J_v9p. Pridobljeno 6.6.2014.
- [33] jsoup. Dostopno na: <http://jsoup.org>. Pridobljeno 18.3.2014.
- [34] HtmlUnit. Dostopno na: <http://htmlunit.sourceforge.net>. Pridobljeno 18.3.2014.
- [35] node.io. Dostopno na: <https://www.npmjs.org/package/node.io>. Pridobljeno 6.6.2014.
- [36] Beautiful Soup. Dostopno na: <http://www.crummy.com/software/BeautifulSoup>. Pridobljeno 6.6.2014.
- [37] Html Agility Pack. Dostopno na: <http://htmlagilitypack.codeplex.com>. Pridobljeno 6.6.2014.
- [38] A. Berglund, S. Boag, D. Chamberlin, M. F. Fernández, M. Kay, J. Robie, J. Siméon. *XML Path Language (XPath) 2.0 W3C Recommendation*, 2007.
- [39] Seznam občin Slovenije, Skupnost občin Slovenije. Dostopno na: <http://www.skupnostobcin.si/?id=414>. Pridobljeno 20.1.2014.
- [40] Seznam vrtcev z enotami, Ministrstvo za izobraževanje znanost in šport. Dostopno na: <https://krka1.mss.edus.si/registriweb/SeznamVrtci.aspx?Enote=1>. Pridobljeno 20.1.2014.

- [41] Seznam osnovnih šol, Ministrstvo za izobraževanje znanost in šport.
Dostopno na:
<https://krka1.mss.edus.si/registriweb/Seznam1.aspx?Seznam=2010>.
Pridobljeno 20.1.2014.
- [42] Seznam srednjih šol, Ministrstvo za izobraževanje znanost in šport.
Dostopno na:
<https://krka1.mss.edus.si/registriweb/Seznam2.aspx?Seznam=3010>.
Pridobljeno 20.1.2014.
- [43] Seznam višjih strokovnih šol, Ministrstvo za izobraževanje znanost in šport. Dostopno na:
<https://krka1.mss.edus.si/registriweb/Seznam1.aspx?Seznam=5040-enote>. Pridobljeno 20.1.2014.
- [44] Seznam članov po dejavnostih, Združenje zdravstvenih zavodov Slovenije. Dostopno na:
https://zdrzz.si/index.php?option=com_content&view=article&id=68&Itemid=105. Pridobljeno 20.1.2014.
- [45] Iskalnik po lekarnah, Lekarniška zbornica Slovenije. Dostopno na:
<http://www.lzs.si/Mre%C5%BEalekarn/Iskalnik/tabid/80/language/sl-SI/Default.aspx>. Pridobljeno 23.4.2014.
- [46] Scheduled Tasks With Cron for Java. Dostopno na:
<https://developers.google.com/appengine/docs/java/config/cron>.
Pridobljeno 6.4.2014.
- [47] Google Spreadsheets API version 3.0. Dokumentacija dostopna na:
<https://developers.google.com/google-apps/spreadsheets/?hl=de-DE&csw=1>. Pridobljeno 27.1.2014.
- [48] Users Java API Overview. Dostopno na:
<https://developers.google.com/appengine/docs/java/users>. Pridobljeno 10.1.2014.

- [49] Markers. Dokumentacija dostopna na:
<https://developers.google.com/maps/documentation/javascript/markers>. Pridobljeno 15.1.2014.
- [50] Spletna stran projekta Map Icons Collection. Dostopno na :
<http://mapicons.nicolasmollet.com>. Pridobljeno 15.1.2014.
- [51] Too Many Markers! Dostopno na:
<https://developers.google.com/maps/articles/toomanymarkers>.
Pridobljeno 11.3.2014.
- [52] Info Windows. Dokumentacija dostopna na:
<https://developers.google.com/maps/documentation/javascript/infowindows>. Pridobljeno 15.1.2014.
- [53] Izvorna koda izdelane aplikacije. Dostopno na:
<https://bitbucket.org/ninazakelj/iskalniknepremicnin>